

SENZOROVÉ SIETE

DIPLOMOVÁ PRÁCA

JOZEF CISKO

**UNIVERZITA KOMENSKÉHO V
BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A
INFORMATIKY
KATEDRA INFORMATIKY**

Študijný odbor: Informatika

Vedúci diplomovej práce: doc. RNDr. Rastislav Královič, PhD.

BRATISLAVA 2006

Čestne prehlasujem, že som diplomovú prácu napísal sám, s pomocou uvedenej literatúry a internetových zdrojov, pod dohľadom vedúceho diplomovej práce.

Bratislava, 2006

Jozef Cisko

Za hodnotné pripomienky a návrhy počas písania práce a za prejavení dôveru chcem vyjadriť vďaku vedúcemu mojej diplomovej práce doc. RNDr. Rastislavovi Královičovi, PhD. Ďalej sa chcem poďakovať Mgr. Miroslave Lemešovej za podporu a pomoc pri štylizácii a gramatickej úprave textu.

Jozef Cisko

Abstrakt

Cisko, Jozef: Sensorové siete. Diplomová práca, Univerzita Komenského. Fakulta matematiky, fyziky a informatiky, Katedra informatiky. Vedúci diplomovej práce: doc. RNDr. Rastislav Královič, PhD. Bratislava : Fakulta matematiky, fyziky a informatiky UK, 2006. 70 s.

Práca prináša ucelený prehľad doposiaľ získaných poznatkov v oblasti bezdrôtových ad hoc sensorových sietí. Predstavuje ich vývoj od historických systémov až po ich súčasné ponímanie a princípy, na ktorých sú sensorové siete založené. Okrem komunikačnej architektúry a vrstvomého referenčného modelu rozoberá aj najdôležitejšie problémy vynárajúce sa pri implementácii takýchto sietí ako sú MAC protokoly, routovanie a broadcastovanie alebo agregácia dát v sieti. Náplňou práce nie je len zhrnutie praktických výsledkov v daných oblastiach komunikačných sietí, ale aj poukázanie na teoretické problémy, ktoré sú v ich pozadí.

Kľúčové slová: sensorové siete, bezdrôtové ad hoc siete, senzory, MAC protokoly, routovanie, broadcastovanie, lokalizácia, agregácia dát

Obsah

Abstrakt	4
Obsah	5
Zoznam obrázkov	7
1 Úvod	8
2 História	10
2.1 Skorý vývoj senzorových sietí	10
2.2 Moderný výskum	10
2.3 Výskum v 21. storočí	12
3 Architektúra	13
3.1 Vrstvový model senzorových sietí	15
3.2 Softvérová podpora senzorových sietí	18
4 Medium access control	21
4.1 Cyklické vypínanie uzlov	23
4.2 Minimálna množina pokrytia siete	24
5 Agregácia dát	28
5.1 Agregácia pomocou stromu	28
5.2 Agregácia vytváraním súhrnov dát	32
6 Routovanie	36
6.1 Geometrické routovanie	37
6.2 Geometrické routovanie v orientovaných grafoch	39
6.3 Štruktúry vhodné pre geometrické routovanie	41
7 Broadcastovanie	46
7.1 Broadcastovanie založené na zaplavovaní	46

<i>OBSAH</i>	6
7.2 Riadené šírenie	48
7.3 Šírenie informácie v mobilných ad hoc sieťach	49
7.4 Kontrola zahltenia siete	50
8 Lokalizácia	54
8.1 Výpočtová zložitosť lokalizácie	54
8.2 Lokalizácia pomocou majákov	55
8.3 Nasadzovanie majákov	59
9 Záver	62
Literatúra	63

Zoznam obrázkov

3.1	Komunikácia v sensorovej sieti	14
3.2	Štruktúra sensorového uzla	15
3.3	Vrstvový model pre sensorové siete	16
4.1	Umiestnenie mriežky v sensorovom poli.	26
5.1	Agregácia v uzle	29
5.2	Kompletný binárny strom pre q-digest	34
6.1	Očíslovanie hrán	40
8.1	Lokalizovanie prienikom	56
8.2	Lokalizovanie dohodou	57
8.3	Aproximácia lokalizovania dohodou	58

Kapitola 1

Úvod

Technológiu sensorových sietí môžeme zaradiť medzi najdôležitejšie technológie 21. storočia. Využitie nájde v mnohých oblastiach a v budúcnosti bude ovplyvňovať život každého z nás. Sensorová sieť pozostáva z veľkého počtu malých, lacných sensorových uzlov nahusto rozmiestnených v okolí alebo priamo v oblasti nášho záujmu, ktorý má skúmať. Ich úlohou je získavať dáta, a to buď periodicky alebo po nastatí určitej udalosti. Každý sensorový uzol má schopnosť spracovávať informácie a môže obsahovať potenciálne niekoľko zabudovaných sensorov. Môže operovať v akustickom, seizmickom, infračervenom, magnetickom alebo inom režime. Obsahuje nejaký typ pamäte a disponuje informáciou o svojej pozícii, či už prostredníctvom globálneho polohovacieho systému (GPS) alebo lokálneho polohovacieho algoritmu.

Vďaka svojej cene a veľkosti môžu byť takéto senzory nasadzované vo veľkom počte na zemi, vznášať sa vo vzduchu, pod vodou, byť nasadené na živé organizmy, stroje, roboty, dopravné prostriedky, budovy, alebo dokonca aj na ľuďoch. Ich pozícia nemusí byť vopred známa alebo predurčená. Pri určitých aplikáciách je nevyhnutné náhodné rozmiestnenie uzlov v nedostupných alebo nebezpečných podmienkach. Ďalšou črtou sensorových sietí je prítomnosť procesora v uzloch. Nasnímané dáta uzly neposielajú priamo, ale najprv prevedú lokálne výpočty a posielajú ďalej už čiastočne spracované údaje.

Máme teda malé, lacné a šikovné zariadenia, schopné snímať svoje okolie, spracovať získané údaje a komunikovať medzi sebou. Takáto technológia ponúka doteraz nevídané možnosti v mnohých oblastiach. Svoje využitie nájde vo vede, zdravotníctve, armáde, priemysle ako i v našom každodennom živote. Teražšie ale aj budúce nasadenie sensorových sietí zahŕňa aplikácie ako vojenské monitorovanie, fyzické zabezpečenie, kontrolu leteckej premávky, sledovanie cestnej premávky, video sledovanie, priemyselnú automatizáciu, distribuovanú robotiku, monitorovanie životného prostredia, migrácie živočíchov, monitorovanie budov a iných dôležitých štruktúr. Systémy v týchto

aplikáciách môžu pozostávať z veľkých alebo miniatúrnych sensorových uzlov a ich siete môžu byť pevné alebo bezdrôtové. Hoci sa pre rôzne aplikácie môžu sensorové siete dosť líšiť, všetky majú spoločné isté technické črty a problémy. Všadeprítomnosť takýchto sietí nám umožní lepšie vnímať a pochopiť svet okolo nás.

Aby sensorové siete mohli plniť vyššie uvedené úlohy, je potrebné vyvinúť pre ne vhodné komunikačné a organizačné protokoly. Naša práca sa venuje práve tejto problematike. Najprv rozoberá stručný prehľad histórie sensorových sietí, po ktorom nasleduje komunikačná architektúra a predstavenie vrstvového modelu pre sensorové siete. Náplňou tretej kapitoly je dizajn MAC protokolov ovplyvnený obmedzeniami sensorových sietí. Nasleduje problematika agregácie dát ako nevyhnutná technika pre zníženie komunikačnej záťaže siete. V piatej a šiestej kapitole sa venujeme tradičným komunikačným technikám routovaniu a broadcastovaniu. Okrem rôznych prístupov rozoberáme aj teoretické pozadie daných problémov súvisiace so špecifickosťou sensorových sietí. Posledná kapitola sa zaoberá problémom určovania polohy v sieti či už u samotných uzlov alebo u pozorovaného objektu.

Kapitola 2

História

Rozvoj sensorových sietí je previazaný s pokrokom v oblastiach snímania, komunikácie a výpočtovej techniky, či už hardvéru, softvéru alebo algoritmov. Pokrok v jednotlivých oblastiach preto poháňa aj výskum v sensorových sieťach. Za ich skorý príklad však môžeme považovať už aj radarovú sieť pre kontrolu leteckej premávky a podobné systémy, ktoré boli vyvinuté pre špeciálne počítače a komunikačné techniky, dávno pred samotným pojmom sensorových sietí [1].

2.1 Skorý vývoj sensorových sietí

Prvá etapa vývoja sensorových sietí začala ako u väčšiny technológií v armáde. Počas studenej vojny bol americkou stranou vyvinutý a nasadený Sound Surveillance System (SOSUS), pozostávajúci z akustických sensorov uložených na dne oceánu v kritických oblastiach, ktorý slúžil na monitorovanie tichých sovietskych ponoriek. Dnes táto sieť slúži na sledovanie pohybu veľrýb [2]. V tomto období boli tiež nasadené radarové systémy pre ochranu územia Spojených štátov a Kanady pred leteckými útokmi. Tieto systémy mali hierarchickú štruktúru pre spracovanie dát. Dochádzalo k nemu postupne po úrovniach, pokiaľ sa informácia o požadovanej udalosti nedostala k používateľovi. V mnohých prípadoch zohrával kľúčovú rolu ľudský operátor. Hoci boli tieto systémy úzko špecializované pre konkrétnu úlohu a konkrétne zariadenia, poskytli niektoré základné techniky pre moderné sensorové siete.

2.2 Moderný výskum

Moderný výskum sa začal okolo roku 1980 s programom Distributed Sensor Networks (DSN) v Defense Advanced Research Projects Agency (DARPA).

V tomto období bol už niekoľko rokov v prevádzke Arpanet, predchodca dnešného Internetu. Riaditeľ Information Processing Techniques Office v DARPA, R. Kahn, ktorý bol aj spoluvýtvorcom TCP/IP protokolov, presadzoval myšlienku rozšírenia prístupu ku komunikácii v Arpanete na senzorové siete. Predpokladala sa sieť priestorovo rozptýlených nízkonákladových senzorových uzlov, ktoré by spolupracovali medzi sebou, pričom by pracovali autonómne. Bola to smelá myšlienka, ktorá sa však kvôli nedostupnosti potrebných technológií pre takúto sieť stala len výzvou do budúcnosti.

Napriek tomu boli vytvorené testovacie siete a vyvinuté techniky pre ich fungovanie. Medzi významné prínosy z tohto obdobia patrí jadro operačného systému Accent [3] vytvorené výskumníkmi na Carnegie Mellon University v Pittsburghu. Bolo orientované na komunikáciu a malo podporiť rozvoj DSN tolerujúcej chyby. Neskôr na jeho základe vznikol operačný systém Mach [4], ktorý mal aj komerčné využitie.

Ďalšími pozoruhodnými výsledkami boli techniky na spracovanie signálu slúžiaceho na sledovanie helikoptér pomocou distribuovaného poľa akustických mikrofónov, ktoré boli vyvinuté v Massachusetts Institute of Technology, Cambridge [5]. V roku 1980 v Advanced Decision Systems, Mountain View, CA, vznikol sledovací algoritmus viacnásobných predpokladov¹, pre riešenie zložitých situácií sledovania, ktoré zahŕňajú vysokú hustotu pozorovaných cieľov, výpadky ich detekcie alebo falošné poplchy. Algoritmus bol upravený pre distribuovanú implementáciu [6]. Prístup pre riešenie zložitých problémov sledovania tohto algoritmu sa používa dodnes.

Úspešne skončila aj demonštrácia senzorovej siete pre sledovanie nízko letiaceho lietadla pomocou akustických mikrofónov [7]. Uzol bol pri nej tvorený nákladným autom s potrebným vybavením a tichým generátorom elektrického prúdu, napojeným na pole mikrofónov usporiadaných do trojuholníka. Z tohto je jasne vidieť ako bolo finančne a technicky náročné zhotoviť funkčnú senzorovú sieť v danom období.

Napriek tomu boli možnosti senzorových sietí rýchlo rozpoznané tvorcami vojenských systémov a stali sa hlavným komponentom pre sieťovo-centrálne vedenie boja² [8]. Sensory a zbrane boli umiestnené na oddelené plošiny, ktoré pracovali nezávisle od seba. Jednotlivé uzly spolupracovali pomocou komunikačnej siete a informácie sa posielali vhodnému strelcovi. Príkladom je systém Cooperative Engagement Capability [9] vyvinutý námorníctvom Spojených štátov.

¹multiple-hypothesis tracking algorithm

²network-centric warfare

2.3 Výskum v 21. storočí

Dnešné pokroky v mikroeletechne, bezdrôtovej komunikácii a dostupnosť energeticky nenáročných procesorov, posunuli senzorové siete k ich pôvodnej myšlienke. DARPA spustila ďalší výskumný program ťažiaci z posledných technických pokrokov, Sensor Information Technology (SensIT) [10]. Tento program sledoval dva ciele. Vytvoriť techniky umožňujúce ad hoc nasadenie veľkého počtu senzorov v dynamickom prostredí a techniky pre sieťové spracovanie informácií.

Terajší výskum sa sústreďí na vytvorenie efektívnych protokolov pre fungovanie bezdrôtovej ad hoc senzorovej siete pozostávajúcej z veľkého počtu uzlov, ktorá by bola odolná voči chybám a dynamicky sa meniacim podmienkam. Na druhej strane existujú projekty na vytvorenie malého a lacného senzorového uzla vhodného pre takúto sieť. Zaujímavý je projekt Smart Dust na University of California, Berkeley [13], ktorý sa snaží o vytvorenie lacného senzorového uzla o veľkosti maximálne 1 mm^2 . Menšie nároky má projekt COTS Dust [14], snažiaci sa poskytnúť funkčný uzol pomocou dnešných technológií, bežne dostupných v obchodoch, ktorého veľkosť nepresiahne štvorcový palec.

Kapitola 3

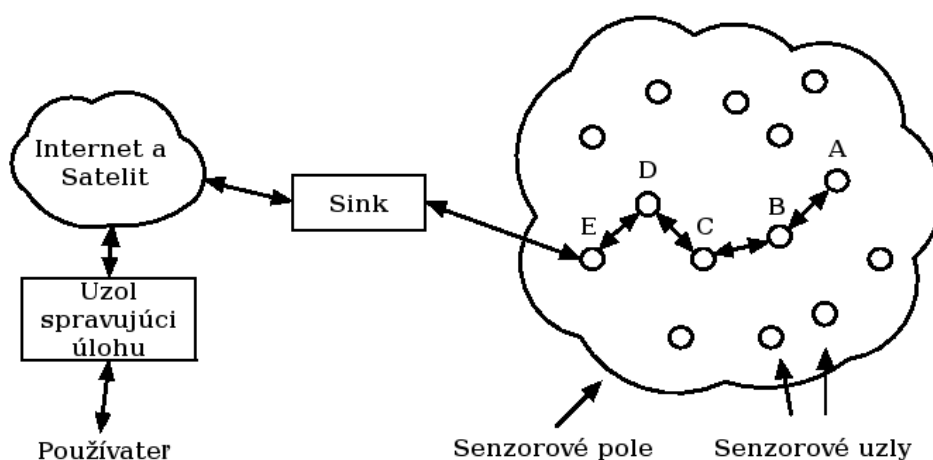
Architektúra

Uzly sensorovej siete sú vo väčšine prípadov náhodne rozmiestnené v oblasti nášho záujmu a vytvárajú takzvané sensorové pole. Každý uzol má schopnosť nasnímať údaje zo svojho okolia a lokálne ich spracovať. Dáta sa potom sieťou prenášajú k špeciálnemu uzlu, ktorý nazývame sink, po hopoch¹ (obrázok 3.1, podľa [11]). Sink s vonkajším svetom komunikuje prostredníctvom Internetu alebo satelitu, na ktorý je pripojený užívateľ cez uzol spravujúci úlohu². Komunikačná architektúra sa môže líšiť v závislosti od toho, na akú aplikáciu je sieť použitá. Základné princípy však ostávajú zachované. Dizajn sensorovej siete ďalej ovplyvňuje viacero faktorov, ktoré sa musia brať do úvahy pri navrhovaní algoritmov a protokolov. Vychádzajú z nich metriky pre určovanie kvality a efektívnosti algoritmov. Najdôležitejšie z nich sú:

- *Tolerancia zlyhania* - kvôli obmedzenému zdroju energie, nepriateľským aktivitám, fyzickému poškodeniu alebo rušeniu prostredím, dochádza k výpadku uzlov v sieti. Tá s nimi musí rátať a zachovať si svoju funkčnosť.
- *Škálovateľnosť* - sensorové siete môžu obsahovať stovky, tisíce a v niektorých prípadoch až milióny uzlov. S tým súvisí aj hustota siete, ktorá výrazne ovplyvňuje efektívnosť komunikačných algoritmov. Navyše sa počet a hustota uzlov v sieti dynamicky mení.
- *Výrobná cena* - s vysokým počtom uzlov prichádzajú aj veľké finančné náklady na tvorbu siete. Preto je dôležité zvoliť uzly s dostatočne nízkou cenou. Funkčnosť, ktorú mali poskytovať chýbajúce komponenty

¹Uzly komunikujú len so svojimi susedmi v dosahu ich vysielača, preto správu preposielajú jednotlivé uzly na ceste od zdroja k cieľu.

²task manager node

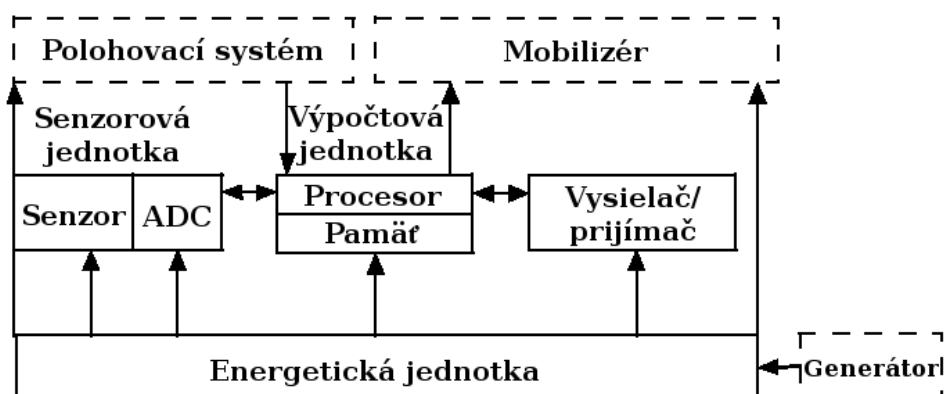


Obrázok 3.1: Komunikácia v senzorovej sieti

ako GPS, dostatok pamäte alebo niektoré senzory, sa musí nahradiť algoritmicke.

- *Hardvérové obmedzenia* - senzorový uzol sa skladá zo štyroch hlavných častí (obrázok 3.2, vybrané z [11]), ktorými sú výpočtová, senzorová, energetická jednotka a vysielač s prijímačom. O snímanie dát sa stará senzorová jednotka, ktorá pozostáva zo senzorov a konvertorov analógového signálu na digitálny (ADC). Po spracovaní dát procesorom, ktorý má zvyčajne k dispozícii nejakú pamäť, sa pripája uzol do siete pomocou vysielača/prijímača a spolupracuje s ostatnými uzlami, o čo sa stará procesor. Okrem týchto komponentov môže uzol obsahovať ďalšie dôležité časti. Niektoré aplikácie vyžadujú premiestňovanie uzlov, ktoré zabezpečuje mobilná jednotka. Väčšina routovacích algoritmov potrebuje pre svoje fungovanie poznať pozíciu uzlov v sieti, čo sa dosahuje algoritmicke, alebo pridaním zariadenia pre určovanie polohy. Jednou z najdôležitejších častí uzla je energetická jednotka, ktorá zásobuje energiou všetky komponenty. Všetky tieto časti senzorového uzla by nemali presahovať určitú veľkosť, obyčajne okolo 1 cm^2 .
- *Topológia siete* - sa dynamicky mení počas celej životnosti siete. Po nasadení do senzorového poľa musia jednotlivé uzly identifikovať svojich susedov, s ktorými budú komunikovať³. Táto informácia sa musí

³Pod topológiou sa tu nemyslí sieť ako celok, jej reprezentácia grafom, kvôli chýbajúcej centrálnej autorite pre jej udržiavanie a nemožnosti použitia globálnych identifikátorov, ale lokálna informácia uzlov o ich susedoch do určitej vzdialenosti.



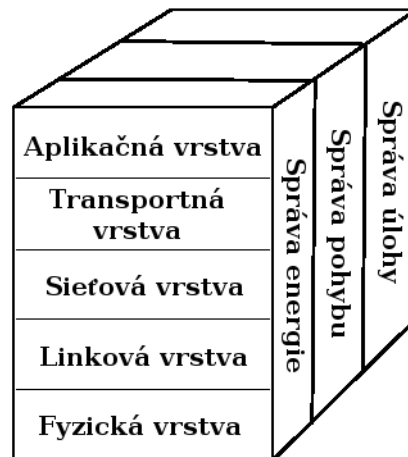
Obrázok 3.2: Štruktúra senzorového uzla

udržiavať aktuálna či už kvôli častým výpadkom uzlov, alebo kvôli nasadeniu nových uzlov z dôvodu nízkej konektivity siete alebo zmeny požiadaviek, ktoré sú na ňu kladené.

- *Prostredie* - použitie senzorových sietí ráta so širokou variabilitou prostredia, čomu sa musí prispôbiť aj jej dizajn. Iné nároky na bezpečnosť má sieť pre detekciu lesných požiarov a iné sieť pre monitorovanie pohybu na nepriateľskom území. Obdobne je rozdiel medzi rušivým vplyvom prostredia v zastavaných a nezastavaných územiach.
- *Vysielacie médium* - do značnej miery ovplyvňuje dizajn siete aj voľba média pre komunikáciu. Najčastejšie sa volí rádio, komunikácia pomocou infračerveného žiarenia alebo optické médium.
- *Spotreba energie* - jednou z najšpecifickejších črt senzorových sietí je neobnoviteľnosť energetických zdrojov, a tým aj jej obmedzená životnosť. Vo väčšine prípadov je nemysliteľné, kvôli nedostupnosti a veľkému počtu, obnovovať batérie senzorových uzlov. Preto je limitované množstvo energie najväčším obmedzením pri tvorbe senzorových sietí a spotreba energie je základnou metrikou pri posudzovaní vhodnosti a efektivity navrhovaných algoritmov.

3.1 Vrstvový model senzorových sietí

Senzorové siete tiež používajú vrstevný model ako poznáme z ISO OSI, alebo TCP/IP referenčného modelu. Jednotlivé vrstvy využívajú služby nižších



Obrázok 3.3: Vrstvový model pre senzorové siete

vrstiev a komunikujú so svojim proťajškom pomocou protokolu prislúchajúceho danej vrstve. Napriek tomu, že model komunikácie je prebratý z klasických sietí, špecifickosť senzorových sietí znemožňuje použitie tradičných bezdrôtových technológií z viacerých dôvodov [11]:

- počet uzlov v senzorovej sieti môže byť niekoľkonásobne väčší ako v tradičnej bezdrôtovej sieti,
- senzorové uzly sú väčšinou rozmiestnené nahusto,
- senzorové uzly sú náchylné na zlyhania,
- topológia siete sa často mení,
- senzorové siete využívajú prevažne broadcastové vysielanie, pričom väčšina ad hoc sietí používa point-to-point komunikáciu,
- senzorové uzly majú obmedzenú výpočtovú silu, pamäť a zdroj energie,
- senzorové uzly nemusia disponovať globálnou identifikáciou kvôli ich veľkému počtu a veľkej hlavičke posielaných správ.

Model predstavený v [11] pozostáva z fyzickej vrstvy, linkovej vrstvy, sieťovej vrstvy, transportnej a aplikačnej vrstvy. Model ďalej obsahuje tri roviny, ktoré sa starajú o správu celej siete a pridelujú uzlom ich úlohy. Je to správa energie, správa pohybu a správa úlohy (obrázok 3.3, podľa [11]).

Jednotlivé vrstvy plnia podobné úlohy ako poznáme z bezdrôtových sietí. Kvôli snahe o čo najmenšie výdaje energie však majú iné priority a inak

pristupujú k riešeniu problémov ako je napríklad strata paketu, kódovanie a podobne.

Fyzická vrstva

Fyzická vrstva je zodpovedná za zvolenie frekvencie, jej generovanie, detekovanie signálu a za moduláciu a šifrovanie. Dôraz je kladený na minimalizáciu spotreby energie, ktorá narastá s komunikačnou vzdialenosťou. Voľba vhodnej modulačnej schémy je tiež dôležitým faktorom pri navrhovaní fyzickej vrstvy. Zatiaľ čo M-árna modulačná schéma umožňuje posielanie viacerých bitov na symbol, binárna modulačná schéma je energeticky efektívnejšia [12].

Linková vrstva

Linková vrstva sa stará o multiplexovanie tokov dát, rozpoznávanie dátových rámcov, o kontrolu chýb a kontrolu prístupu k médiu. Zabezpečuje spoľahlivé point-to-point alebo point-to-multipoint spojenie v sieti a jej samoorganizáciu⁴. Najdôležitejšou časťou linkovej vrstvy je vhodný Medium Access Control protokol (MAC), ktorý má za úlohu vytvoriť infraštruktúru siete, linky medzi uzlami, po ktorých bude prebiehať komunikácia a efektívne zdieľať komunikačné zdroje medzi sensorovými uzlami. Vhodná voľba kontroly chýb môže tiež výrazne ovplyvniť spotrebu energie. Pokiaľ prepisovanie paketov vyžaduje energeticky najnáročnejšie operácie vysielania a prijímania, samoopravné kódy znižujú svojou zložitosťou limitovanú výpočtovú silu uzla.

Sieťová vrstva

Sieťová vrstva zabezpečuje spojenie medzi ľubovoľnými uzlami siete, respektíve medzi uzlami a prístupovými bodmi, sink-mi. Bežné routovacie alebo broadcastovacie algoritmy opäť neprichádzajú do úvahy kvôli veľkosti siete a limitovanej energii. Na tejto úrovni sa často od protokolov vyžaduje aj agregácia dát. Pre užívateľa často nie je nevyhnutné poznať presne namerané údaje od všetkých uzlov, skôr má záujem o merania v danej oblasti. Preto sa dáta po ceste k užívateľovi agregujú, čím sa znižuje objem prenášaných dát a šetrí sa energia. S tým súvisí aj adresácia dát hodnotami, keď požiadavka na sieť nemá tvar “Aké teploty boli namerané v daných oblastiach? ”, ale “V ktorých oblastiach presiahla teplota 40 stupňov? ”.

⁴Schopnosť siete vysporiadať sa s meniacimi podmienkami, výpadkami uzlov, a tým aj zmenou jej infraštruktúry.

Transportná vrstva

Podľa I. F. Akyildiza a kol. [11] je táto vrstva potrebná pri prístupoch na Internet alebo do iných externých sietí. Inak sa jej nevenuje väčšia pozornosť. Na jej používanie postačuje svojimi vlastnosťami aj TCP protokol, ktorý sa používa pri komunikácii medzi sink-om a užívateľom cez Internet a UDP protokol na komunikáciu medzi sink-om a ostatnými uzlami. Tieto protokoly však nie sú preberané bez zmeny. Sensorové siete nepoužívajú globálnu adresáciu, preto je potrebná ich úprava kvôli adresácii pomocou hodnôt a zohľadňovaní energetickej náročnosti.

Aplikačná vrstva

Pre sensorové siete existuje veľké množstvo definovaných a navrhovaných aplikácií. Táto vrstva a jej protokoly je však z veľkej časti nepreskúmaná.

Správa energie, pohybu a úlohy

Okrem vyššie spomínaných vrstiev dopĺňajú model ešte správa energie, správa pohybu a správa úlohy. Správa energie sa stará o spotrebu energie v sieti, o jej minimalizovanie, a tým predĺženie životnosti siete. To dosahuje napríklad vypínaním prijímača uzla po prijatí správy a jej spracovaní, aby sa predišlo prijímaniu duplicitných správ. Pri nízkom stave energie musí uzol svojim susedom oznámiť, že sa už nemôže podieľať na preposielaní správ. Správa pohybu sleduje pohyb uzlov v sieti, aby sa zabezpečilo routovanie k užívateľovi a prehľad o susedoch jednotlivých uzlov. Správa úlohy plánuje a rovnomerne rozdeľuje medzi uzly snímanie, pretože nie je potrebné, aby všetky uzly snímali dáta naraz. Pre snímanie sa vyberajú uzly s najväčšími zdrojmi energie. Správy týmto zabezpečujú spoluprácu medzi uzlami a efektívne využívanie zdrojov.

3.2 Softvérová podpora sensorových sietí

Hoci už existujú experimentálne siete a uzly, ktoré spĺňajú pôvodnú predstavu sensorových sietí ako je napríklad COTS Dust [14] alebo Motes uzly [15], stále nie sú k dispozícii reálne siete obsahujúce tisíce uzlov, na ktorých by sa dalo zistiť správanie a vlastnosti navrhovaných protokolov a algoritmov. Z tohto dôvodu sa vyvíjajú rôzne modely bezdrôtových ad hoc sensorových sietí, ktoré by umožnili simulácie takýchto algoritmov. V. K. Prasanna nastolil otázky pre bádanie v oblasti sensorových sietí vedúce k vytvoreniu ich abstraktného modelu [16]:

- Aké sú vhodné abstraktné modely pre systémy sensorových sietí, ktoré môžu použiť programátori pre vývoj aplikácií, ktoré su efektívne s ohľadom na relevantné metriky výkonnosti?
- Ako navrhnuť a optimalizovať algoritmy tak, aby mohla byť realizovaná optimálna alebo skoro optimálna časová a energetická náročnosť pri existencii vhodného nástroja pre kontrolu nad nižšími vrstvami protokolov a hardvéru ako rádiový dosah, energetické stavy a podobne?
- Aké sú znova použiteľné primitíva pre rapídne zlučovanie aplikácií?

Niektoré z algoritmických a teoretických problémov motivovaných sensorovými sieťami ako i nové metriky efektívnosti a kvality algoritmov prezentuje P. G. Spirakis v [21].

Výskum sa sústreďí na nájdenie vhodných protokolov a algoritmov pre sensorové siete na jednej strane a dizajnu hardvéru na strane druhej. Jednou z kľúčových technológií, ktorým sa venuje málo pozornosti, je softvérová podpora pre správu a ovládanie sensorových zariadení. J. Hill a kol. z týchto dôvodov vytvorili systémovú architektúru a operačný systém pre sensorické siete, založenú na dnes dostupných technológiách, TinyOS (Tiny microthreading operating system) [22]. Hlavnými cieľmi tohto operačného systému pre sensorové siete bolo:

- efektívne využívanie hardvéru,
- dosiahnutie efektívnej modularity a robustnosti,
- podpora súbežne prebiehajúcich operácií.

Napriek tomu, že sa im nepodarilo splniť pôvodné ciele, vytvorili operačný systém s malou náročnosťou na fyzickú pamäť (celej aplikácii postačuje len 226 bytov), podporou súbežne prebiehajúcich operácií (avšak s výraznými režijnými nákladmi), efektívnou modularitou a limitovaným fyzickým paralelizmom, ktorý má rôznorodú použiteľnosť.

Väčšina experimentálnych sensorových sietí, ktorými sa testujú vlastnosti a výkonnosť navrhovaných protokolov a algoritmov využíva práve operačný systém TinyOS.

Ďalším problémom, ktorému sa venuje málo pozornosti, je bezpečnosť v sensorových sieťach. Pritom pre niektoré aplikácie je bezpečnosť kritickou vlastnosťou. Príkladom môže byť armádna sieť nasadená na monitorovanie pohybu nepriateľských jednotiek. Je nevyhnutné, aby bola zabezpečená ako proti úniku, tak aj proti podstrčeniu nepravých údajov. Všeobecný prístup pre tvorbu protokolov zameraný výhradne na šetrenie energie však nemôže

viest k architektúre s dobrou bezpečnosťou. Preto je potrebné zvoliť správny pomer medzi spotrebovanou energiou, použitou pamäťou a bezpečnostnou robustnosťou [23] v distribuovanej sensorovej sieti. Bežné bezpečnostné protokoly pridávajú okolo 16 - 32 bytov do hlavičky správ. Pri limitovanej pamäti, nízkemu výkonu procesorov, obmedzeným energetickým zdrojom a paketmi o veľkosti približne 30 bytov si sensorové siete takéto protokoly nemôžu dovoliť. Naopak si vyžadujú protokol s minimálnym dopadom na výkon, ľahkou implementáciou a energetickou nenáročnosťou. Jedným z riešení tohto problému je TinySec [24] bezpečnostná architektúra pre linkovú vrstvu sensorových sietí, ktorá bola plne implementovaná a začlenená do TinyOS. Je založená na bezpečnostných primitívach, ktoré už boli dokázané ako bezpečné a využíva obmedzenia sensorových sietí vo svoj prospech. Nízkou šírkou pásma je limitovaný aj počet paketov, ktoré môže útočník odpočuť alebo do komunikácie vložiť.

Pre efektívne používanie sensorových uzlov je potrebná taktiež softvérová podpora pre ukladanie dát. Niektoré aplikácie umožňujú len sporadické posielanie dát. S problémom dočasného uloženia údajov sa môžeme stretnúť aj pri aplikáciách s častým prenosom dát, v ktorých je spôsobený vplyvom hardvérového zlyhania alebo rušenia prostredím. Riešením by bol vhodný súborový systém pre sensorové siete. Jeho dizajn sa však musí vysporiadať s obmedzeniami ako je limitovanie operačnej pamäte, trvalého úložného priestoru a energie. Veľa sensorových uzlov používa ako úložné jednotky flash pamäte, ktoré majú obmedzený počet zápisov. Pre takéto mikrosensorové uzly vytvorili H. Dai a kol. zapisovaciu štruktúru ELF [25]. Takýto súborový systém taktiež uľahčí prácu aplikáciám pracujúcim so súbormi ako je TinyDB [26] alebo Cougar [27], určenými pre agregáciu dát abstrakciou základných operácií so súbormi ako je *open*, *read*, *write* a *delete*.

Kapitola 4

Medium access control

Podobne ako pri iných sieťach, ktoré zdieľajú jedno médium, je aj v senzorových sieťach dôležitým prvkom, ktorý umožňuje ich prevádzku, kontrola prístupu k médiu. Pre bezdrôtové siete existuje veľké množstvo MAC protokolov, ktoré sú kvôli obmedzeniam v senzorových sieťach neefektívne alebo rovno nepoužiteľné. Ako príklad môžeme uviesť time-division multiple access (TDMA) [17], code-division multiple access (CDMA) [18] a IEEE 802.11 [19]. Zvýšený záujem o senzorové siete viedol k vytvoreniu pracovnej skupiny v IEEE štandardizačnej komisii a k vytvoreniu nového štandardu 802.15.4 s MAC protokolom a fyzickou vrstvou s nízkou spotrebou energie [20].

Základnými úlohami MAC protokolu je vytvoriť a udržiavať infraštruktúru siete a zabráňovať kolíziám medzi interferujúcimi uzlami. Dôležitou požiadavkou na protokol je šetrenie energetickými zdrojmi senzorovej siete. Existujúce návrhy MAC protokolov majú rôzne motivácie svojho vzniku a snažia sa o dosiahnutie efektivity a energetickej nenáročnosti rozdielnymi prístupmi. Väčšinou sú navrhované a optimalizované pre konkrétny typ aplikácie, pracovné podmienky a predpoklady. Zameriavajú sa zväčša len na jeden z faktorov spôsobujúcich plytvanie energetickými zdrojmi siete a snažia sa o jeho minimalizáciu. Tieto faktory však v rôznych podmienkach a pri rôznych druhoch úloh majú rozdielne miery vplyvu na spotrebu energie.

Kvalita bezdrôtovej komunikácie závisí od prostredia, od použitej frekvencie, od konkrétnej modulačnej schémy a prípadne aj od použitého komunikačného média. V senzorových sieťach je ich vplyv na kvalitu ešte vyšší ako u bezdrôtových sietí so základnými stanicami. Prvé kvantitatívne merania doručovania paketov uskutočnili J. Zhao a R. Govindan [28] s prekvapivými zisteniami. Merania vykonávali na fyzickej vrstve a MAC protokoloch v troch rôznych prostrediach. Umiestnili sieť s jednoduchou lineárnou topológiou do administratívnej budovy, prirodzeného mierne zalesneného prostredia a na nekryté parkovisko. V interiéroch až polovica liniek dosahovala 10% stratu

paketov a až tretina liniek viac ako 30% stratu. Túto variabilitu vysvetľujú existenciou šedej oblasti v komunikačnom dosahu uzla. Prijímatelia v tejto oblasti dostávajú poškodené pakety, pričom v niektorých prostrediach môže tvoriť až tretinu komunikačného pokrytia uzla. MAC protokoly dopadli ešte horšie. Približne 50% až 80% energie sa spotrebuje na prekonanie kolízií paketov a rušivých vplyvov prostredia.

Prostredie a podmienky, v ktorých sieť operuje, teda výrazne ovplyvňujú jej výkonnosť a spotrebu energie. Vznikla tak potreba modelovania energetickej náročnosti v senzorových sieťach. M. Achir a L. Ouvry [29] vypracovali model pre odhadovanie energetických nárokov pre jednotlivé uzly v senzorovej sieti. Dosahujú to použitím modelovania CSMA/CA mechanizmov použitých v IEEE 802.15.4 štandarde pomocou Markovovho reťazca. Berú pritom do úvahy ako MAC protokol, tak aj fyzickú vrstvu. Začínajúc z globálnych parametrov, buď predpokladaných alebo definovaných štandardom, vypočítavajú pravdepodobnosti prechodu Markovovho reťazca pri použití ich pravdepodobnostného modelu pre interferenciu a komunikačnú premávku. Odhadovaná spotreba energie pre senzorový uzol je potom získaná ako dôsledok konvergencie Markovovho reťazca.

W. Ye a kol. [30] za hlavné atribúty dobrého MAC protokolu považujú energetickú efektivitu a adaptívnosť na zmeny veľkosti siete, hustoty uzlov a topológie. Ostatné typicky dôležité atribúty ako je spravodlivosť¹, doba odozvy, priechodnosť a priepustnosť považujú za druhoradé. Ako hlavné príčiny plytvania energiou identifikujú:

- *Kolízie* - poškodený paket musí byť zahodený a následne opäť preposlaný. To nielen mýňa energiu uzla, ale aj zvyšuje dobu odozvy siete.
- *Nadmerné počúvanie* - uzol preberá aj pakety, ktoré sú určené pre iné uzly.
- *Náklady na prevádzku protokolu* - posielanie a prijímanie kontrolných paketov a/alebo hlavičky správ zafažujú komunikáciu, a tým aj energetické zdroje siete.
- *Aktívne počúvanie* - počúvanie za účelom zachytenia komunikácie, ktorá však nebola vyslaná. Ak nenasnímajú žiadne dáta, uzly sú väčšinu času v tomto stave. Protokoly ako IEEE 802.11 alebo CDMA vyžadujú aktívne počúvanie, ktoré podľa meraní pohltí 50% až 100% energie pre prijímanie správ.

¹Rovnaká možnosť prístupu k zdrojom siete pre všetky uzly.

Navrhované MAC protokoly sa snažia eliminovať tieto faktory. Keďže najväčšie plytvanie energetickými zdrojmi v sieti nastáva pri aktívnom počúvaní, väčšinou sa sústredia na odstránenie tohto problému. Prirodzeným riešením aktívneho počúvania je vypínanie uzlov, ak v sieti neprebíha komunikácia, alebo vypínanie ich komunikačnej jednotky počas spracovávania údajov.

4.1 Cyklické vypínanie uzlov

Jeden z prístupov pre riešenie problému aktívneho počúvania je založený na cyklickom vypínaní uzlov. Príkladom je S-MAC protokol [30]. Eliminuje faktory spôsobujúce plytvanie energiou na úkor výkonu. Dosahuje to dvoma spôsobmi. Prvým je zavedenie takzvaných cyklov krátkej služby². Uzly sa periodicky dostávajú do spiaceho stavu, kedy majú vypnutý vysielateľ/prijímač. Podobnú schopnosť má aj IEEE 802.11 protokol (nazývanú energiou šetriaci mód), ktorá sa aktivuje po dlhšom čase aktívneho počúvania. U S-MAC protokolu je však cyklus krátkej služby jednou zo základných operácií na uzloch, kedy každý uzol striktné dodržiava svoj spací rozvrh a používa sa koordinované spanie medzi susednými uzlami. Periodickým striedaním aktívneho a spiaceho stavu uzlov sa však zvyšuje doba odozvy, kedy vysielací uzol musí čakať na prijímateľa, ktorý je v spiacom stave. Na zmiernenie tohto dôsledku sa používa adaptívne počúvanie.

Základnou myšlienkou je povoliť uzlu, ktorý zachytil vysielanie určené pre svojho suseda, prebudiť sa na krátky čas na konci tohto vysielania. Vďaka tomu, ak má prebudený uzol vykonať ďalší hop v ceste správy, je okamžite pripravený na jej prijatie od svojho suseda. Ak nie, v krátkom časovom intervale neprijme žiadnu správu a môže sa vrátiť do spiaceho stavu. Uzly majú teda možnosť prepínať sa medzi aktívnym a spiacim stavom v závislosti od premávky v sieti.

Na redukciu kontrolnej hlavičky a aktívneho počúvania slúži aj rozdelenie veľkej správy do menších paketov, ktoré sú potom naraz odoslané. Znižuje sa tým spravodlivosť, pretože jeden uzol má dlhší čas prístup k médiu, ale šetrí sa energia. Využitím spomínaných techník dosahuje S-MAC protokol významné šetrenie energie v sieti, a tým aj predĺženie jej životnosti oproti IEEE 802.11 protokolu, čo dokazujú W. Ye a kol. experimentálnymi výsledkami implementovaním na Motes uzly pri použití TinyOS.

Rozdielny prístup k striedaniu aktívneho a spiaceho stavu uzlov používa WiseMAC protokol (Wireless sensor MAC), ktorý predstavili A. El-Hoiydim a kol. v [31]. Nepoužíva sa striktné striedanie stavov ako u S-MAC protokolu, ale sleduje aktivitu na médiu a podľa nej uzly vstupujú do spiaceho

²low-duty-cycle

stavu. Uzol prevádza na komunikačnom kanáli vzorkovanie, teda počúva komunikáciu počas krátkeho časového úseku³. Všetky uzly vzorkujú s rovnakou konštantnou periódou, ale s nezávislými posunmi vzorkovacích rozvrhov. Uzol takto pri detekcii komunikácie na kanáli ostáva aktívny, pokiaľ neprijme dátový rámec alebo sa kanál opäť stane nečinným.

Pri vysielaní správy sa pred ňu pridáva hlavička pre prebudenie uzla, s veľkosťou rovnou vzorkovacej perióde, aby sa zaručilo jej prijatie. Takáto technika zaručuje veľmi nízke spotreby energie pri nečinnom kanáli. Jej nevýhodou je však zaťažovanie siete hlavičkami pre prebudenie uzlov, ktoré treba minimalizovať.

WiseMAC to dosahuje jednoduchou myšlienkou zistenia vzorkovacích rozvrhov priamych susedov. Do potvrdzovacích správ o prijatí paketu sa vkladá informácia o zostávajúcom čase do najbližšej vzorky. Uzol si tak uchováva aktuálny posun vo vzorkovacích rozvrhoch u uzlov, s ktorými najčastejšie komunikuje. Túto informáciu využíva na správne načasovanie posielania správ, čím minimalizuje hlavičku pre prebúdzanie. Pri porovnaníach s S-MAC protokolom autori uvádzajú až sedemnásobné predĺženie životnosti siete použitím WiseMAC protokolu oproti S-MAC s 10% cyklami.

Z oboch doteraz predstavených prístupov pre fungovanie MAC protokolu vychádza B-MAC [32]. Protokol vytvára služobné cykly pre uzol periodickým vzorkovaním kanála, ktorý sa nazýva nízko energetické počúvanie⁴. Vždy po prebudení uzol zapne prijímač a skontroluje prebiehajúcu aktivitu na prenosovom kanáli. Ak nejakú detekuje, uzol ostáva aktívny pokiaľ neprijme prichádzajúci paket. Po jeho prijatí prechádza opäť do spiaceho stavu. Ak uzol nedetekuje žiadny prichádzajúci paket, po určitom časovom intervale je prinútený vrátiť sa do spiaceho stavu.

Vypínanie uzlov v sieti výrazne šetrí energiu, a tým aj predlžuje životnosť siete. Tento prístup však má na druhej strane aj svoje negatíva. Zväčšuje sa čas odozvy pri čakaní na spiace uzly, protokoly nie sú spravodlivé a zväčšujú veľkosť správ pridávaním organizačných informácií do ich hlavičiek.

4.2 Minimálna množina pokrytia siete

Vypínanie určitých uzlov sa javí ako nevyhnutná technika pri senzorových sieťach, ktorá zabezpečuje jej dlhšiu životnosť. Vo všeobecnosti je rýchlosť pozorovaného objektu alebo udalosti menšia ako rýchlosť šírenia signálu. Vďaka tejto vlastnosti protokoly ako sú S-MAC, WiseMAC alebo B-MAC znižujú energetickú spotrebu siete na úkor jej reakčnej doby. Tento prístup

³Napríklad počas trvania jedného modulačného symbolu.

⁴low power listening

však nie je vhodný pre všetky aplikácie. Niektoré sú veľmi citlivé na dobu odozvy siete.

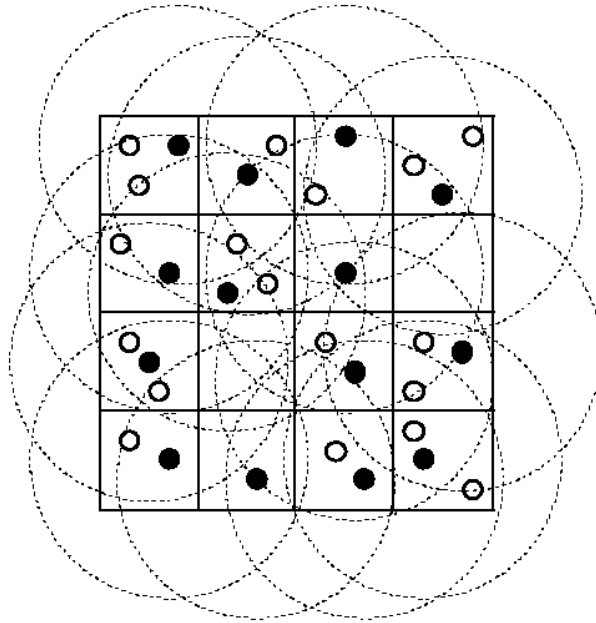
Senzorové uzly sa v mnohých prípadoch nasadzujú vo veľkom počte, pričom dosahujú vysokú hustotu (okolo $20 \text{ uzlov}/m^3$). V takejto sieti s veľkou hustotou je vysoko pravdepodobné, že senzorové pokrytie jedného uzla je pokryté aj jeho susedmi. Navyše môže dochádzať k častým kolíziám a interferenciám pri simultánnom vysielaní susedných uzlov, ktoré budú posielať rovnaké nasnímané dáta, čo vedie k zbytočnému plytvaniu energiou. Riešením je kontrola hustoty, kedy sa ako aktívne udržiavajú len niektoré uzly tak, aby bolo zabezpečené kompletne pokrytie siete. Keď niektorý z týchto uzlov vyplytvá svoje zdroje, neaktívne uzly ho môžu nahradiť, pričom musia byť dodržané nasledovné podmienky:

- udržanie senzorového pokrytia originálnej siete,
- udržanie konektivity siete.

Ak je komunikačný dosah uzlov aspoň dvakrát väčší ako ich senzorový dosah, kompletne senzorové pokrytie konvexnej oblasti implikuje aj konektivitu pracujúcich uzlov v nej [33]. Za tohto predpokladu sa problém redukuje na nájdenie minimálnej podmnožiny uzlov, ktorá ešte zabezpečí senzorové pokrytie ako mala originálna sieť.

Distribuovaný, lokálny algoritmus pre kontrolu hustoty riešiaci tento problém predstavili J. Jiang a W. Dou v [34]. Počas fungovania siete sa v nej striedajú dve fázy, *plánovacia* a *snímacia*. Kvôli zníženiu komunikačných nákladov na plánovaciu fázu by mala byť snímacia dostatočne dlhá v porovnaní s plánovacou. Tá je ďalej rozdelená na *fázu zisťovania susedov* a *vyhodnocovacu fázu*. Na začiatku fázy pre zisťovanie susedov každý uzol broadcastuje *hello* správu a nastaví si časovač. Po jeho vypršaní disponuje uzol informáciou o svojich susedoch vzdialených na jeden hop a je schopný vytvoriť si množinu susedov a efektívnu množinu susedov. Prechádza tak do vyhodnocovacej fázy, v ktorej vyhodnocuje algoritmus pre kontrolu hustoty a na základe výsledkov sa rozhodne pre aktívny alebo neaktívny stav, čo oznámi susedom poslaním správy *off*.

Problémom sú slepé miesta, ktoré vznikajú ak sa susedné uzly dostanú do vyhodnocovacej fázy simultánne a oba sa rozhodnú pre neaktívny stav. Riešením je náhodná schéma umožňujúca sekvenčné vyhodnocovanie susedných uzlov. Na začiatku vyhodnocovacej fázy každý uzol počká náhodný čas. Uzly si pritom upravujú efektívne množiny susedov podľa prijatých *off* správ. Táto schéma však tiež nie je dostačujúca. Preto ešte každý z neaktívnych uzlov čaká istý čas a keď počas neho nedostane žiadne správy *off*, definitívne vypne svoju komunikačnú jednotku.



Obrázok 4.1: Umiestnenie mriežky v senzorum poli. Tmavé uzly sú aktívne, čiarkované kružnice predstavujú dosah ich senzorov.

Pozrime sa teraz z druhej strany na problém vybratia minimálnej množiny uzlov, ktorá senzorum pokrýva skúmanú oblasť. Nebudeme vypínať uzly, ktoré nenarušujú pokrytie, ale budeme vytvárať množinu uzlov pokrývajúcu celú oblasť. Problém senzorum pokrytia siete týmto prístupom rieši jednoduchý greedy algoritmus⁵ pre pokrývanie sektorov [35]. Funguje v dvoch krokoch:

- skontroluje, či senzory pokrývajú oblasť záujmu a vyhlási chybu, ak sa nedá dosiahnuť vyhovujúce riešenie,
- v každom kroku si vyberie senzorum uzol, ktorý pokrýva maximálny počet sektorov.

Jednoduchým riešením je aj pokrytie senzorum poľa mriežkou. Nech R_s je dosah senzorum uzla. Zvoľme veľkosť bunky v mriežke $\frac{R_s}{\sqrt{2}} \times \frac{R_s}{\sqrt{2}}$. Preložením senzorum poľa takouto mriežkou získame jeho senzorum pokrytie zvolením jedného uzla v každej bunke. Pri predpoklade dvojnásobného komunikačného dosahu uzla budú navyše uzly schopné komunikácie s reprezentantmi v

⁵pažravý algoritmus - jedna z techník pre tvorbu efektívnych algoritmov. Algoritmus buduje riešenie v krokoch, pričom si v každom z nich vyberá lokálne najlepšiu možnosť vedúcu k cieľu. Tento prístup však nezaručuje dosiahnutie optimálneho riešenia.

susedných bunkách mriežky (obrázok 4.1). Problém nastane ak sa v niektorej bunke nenachádza žiadny uzol. Pre minimalizovanie prázdnych buniek je potrebná vhodná voľba pozície mriežky, ktorá sa označuje ako problém umiestnenia mriežky⁶ [36].

Vyššie spomínané prístupy nie sú jedinými, ktoré sa používajú pri navrhovaní MAC protokolov. Niektoré návrhy sledujú iné ciele ako je znižovanie spotreby energie, alebo sú určené pre špecifické aplikácie, ktorých kľúčové vlastnosti sa musia dosiahnuť aj na úkor ich životnosti. Príkladom môže byť Rate-adaptive MAC protokol [37] postavený na simultánnom vysielaní a prispôbovaní kódovania kanálu interferencii pre zlepšenie priepustnosti siete, UAMAC⁷ protokol pre heterogénne siete [38] alebo protokoly vyžadujúce synchronizáciu potrebnú pre presné určovanie času nastatia udalosti [39]. Tieto protokoly sú však zamerané na konkrétnu aplikáciu a špecifické podmienky a neprispievajú tak k riešeniu všeobecného problému MAC protokolu u senzorových sietí. Najvýraznejším rozdielom medzi bezdrôtovými a senzorovými sieťami je neobnoviteľnosť energetických zdrojov, a preto by hlavnou motiváciou pri dizajne MAC protokolov senzorových sietí malo byť ich efektívne využívanie.

⁶grid placement problem

⁷Unidirectional-link aware MAC protocol

Kapitola 5

Agregácia dát

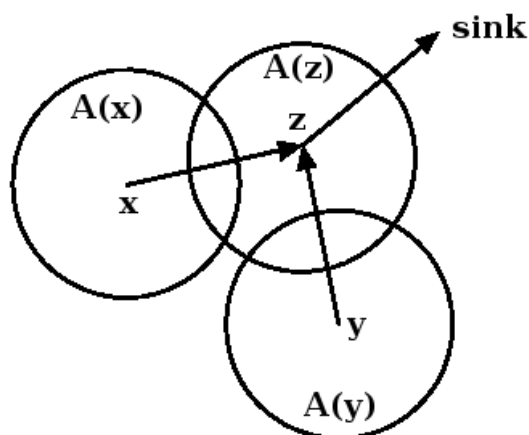
Senzorové siete sa vyznačujú relatívne vysokou hustotou nasadených uzlov. Dáta, ktoré snímajú susedné alebo blízko pri sebe ležiace uzly tak vykazujú určitú mieru podobnosti, geograficky spolu súvisia. V mnohých prípadoch nie je potrebné poznať presné dáta namerané každým uzlom, ale dáta namerané uzlami v danej oblasti. Pre užívateľa je vhodnejšie dotazovať sieť ako v prípade databáz (Cougar, TinyDB), teda zistiť hodnoty ako sú *Min*, *Max*, *Avg*, *Sum* a podobne.

Tieto vlastnosti sensorových sietí umožňujú agregáciu dát, čím sa výrazne znižuje objem prenášaných informácií, teda operácií, ktoré sú energeticky najnáročnejšie. Pri agregovaní odstraňujeme duplicitné dáta (vznikajú pri prekrývaní oblastí, ktoré snímajú jednotlivé uzly) alebo pri výpočtoch priemerných hodnôt alebo sumácie prenášame cez uzly už čiastočné výsledky, aby sme minimalizovali množstvo informácií preposielaných sieťou. Keďže poslanie jedného bitu je rádovo až trikrát drahšie ako vykonanie jednoduchej operácie v zmysle spotreby energie, je myšlienka spracovania dát a aproximácia výsledku správnym krokom. Agregácia dát sa tak stáva jednou z kľúčových techník v ad hoc distribuovaných sensorových sieťach. Agregáciu schému nazývame *oportunistická*¹, ak sa dáta k centrálnemu uzlu posielajú po najkratších cestách, teda agregáciou nie sú spôsobené žiadne extra routovacie náklady.

5.1 Agregácia pomocou stromu

Predpokladajme, že veľkosť nasnímaných dát je úmerná veľkosti oblasti, ktorú senzor sníma. Potom veľkosť zbytočných dát posielaných k sink-u je rovná veľkosti prekrývajúcich sa dosahov jednotlivých senzorov. Tieto zbytočné

¹opportunistic



Obrázok 5.1: Agregácia v uzle

Veľkosť snímanej oblasti uzlom x je $A(x)$ a nasnímané dáta sú úmerné tejto oblasti pre nejakú konštantu γ , teda veľkosti $\gamma|A(x)|$. Uzly x a y posielajú svoje nasnímané dáta uzlu z , ktorý posieľa smerom k sink-u skombinované dáta z uzlov x , y a z . Celková veľkosť agregovaných dát pri uzle z je $\gamma|A(x) \cup A(y) \cup A(z)|$ a celkové komunikačné náklady $\gamma(|A(x)| + |A(y)| + |A(x) \cup A(y) \cup A(z)|)$.

dáta môžu byť odstránené alebo komprimované rôznymi kódovacími technikami [40].

Pretože samotné sensorové uzly sú schopné agregácie dát², môžeme zostrojiť agregáčny strom zakorenený v sink-u (uzle, ktorý dotazuje dáta), po ktorého hranách sa budú informácie prenášať smerom ku koreňu, pričom sa budú v uzloch odstraňovať zbytočné informácie. Informácie sa spracovávajú po úrovniach. Uzol najprv prijme dáta od všetkých svojich detí v strome, spracuje ich a pošle svojmu otcovi. Tvar tohto stromu vplýva na to, ako skoro a ako efektívne sú prebytočné dáta odstránené a aké budú komunikačné náklady [41]. Čím hlbšie sa senzor v strome nachádza, tým skôr sa jeho dáta budú agregovať.

Uvažujme o idealizovanom modeli sensorovej siete, kde sú uzly usporiadané do mriežky veľkosti $N \times N$ s centrálnym výpočtovým uzlom na pozícii $(0, 0)$. Nech každý uzol môže komunikovať len so svojimi štyrmi susedmi na mriežke. Pre modelovanie geografickej súvislosti³, nech každý senzor môže snímať dáta zo štvorca o veľkosti $\frac{k}{2} \times \frac{k}{2}$ (alebo z kruhu s polomerom $\frac{k}{2}$) so stredom v uzle. K budeme označovať ako parameter súvislosti. Nech množina $A(x)$ označuje oblasť snímanú senzorom x . Ak budeme agregovať dáta

²disponujú výpočtovou jednotkou

³correlation

z množiny sensorov S , potom veľkosť výslednej komprimovanej informácie je $I(S) = \left| \bigcup_{x \in S} A(x) \right|$, teda veľkosť celej oblasti pokrytej senzormi v S [42].

Zjednodušený príklad ukazuje obrázok 5.1 (podľa [41]). Parameter k často závisí od intenzity nasnímaných dát⁴. Hoci sa jedná o veľmi zjednodušený model s veľmi silnými predpokladmi, často sa používa k štúdiu spracovávaní informácií v sensorových sieťach a pri niektorých scenároch sa dá priamo aplikovať.

Pre uzol (i, j) budeme nazývať uzly $(i - 1, j)$ a $(i, j - 1)$ dolní susedia a uzly $(i + 1, j)$ a $(i, j + 1)$ horní susedia. Teraz vybudujeme strom po ktorom budú dáta prenášané k centrálnemu uzlu, pričom budú po ceste agregované. Keďže chceme dosiahnuť oportunistickú agregáciu, každý uzol má na výber len dve možnosti spomedzi svojich dolných susedov. Agregácia bude prebiehať nasledovne:

- uzol počká pokiaľ obaja jeho horní susedia odošlú dáta (nie nutne jednému),
- agreguje údaje, ktoré nasnímal lokálne, so všetkými dátami, ktoré prijal od horných susedov a odošle ich jednému z dolných susedov.

Enachescu a kol. [42] navrhujú jednoduchý znáhodnený algoritmus pre voľbu uzla, ktorému budú dáta odoslané. Uzol (i, j) si zvolí svojho ľavého suseda s pravdepodobnosťou $\frac{i}{i+j}$ a spodného suseda s pravdepodobnosťou $\frac{j}{i+j}$. V takejto schéme sú všetky najkratšie cesty z uzla (i, j) do uzla $(0, 0)$ vyberané s rovnakou pravdepodobnosťou a algoritmus má konštantný aproximačný faktor k optimálnemu agregáčnemu stromu, a to simultánne pre všetky parametre súvislosti k ⁵. Vďaka jednoduchosti je algoritmus vhodný pre sensorové siete, ale jeho výsledky platia len pre celkovú cenu stromu⁶ a kriticky závisí od predpokladu, že informácie sú po sieti distribuované rovnomerne.

Pre vytváranie agregáčnych stromov existuje mnoho aproximačných heuristik. Spomeňme aspoň niektoré [41]:

- *Minimálna vzdialenosť k ďalšiemu hopu (MinDist)* - volí sa najbližší sused (blízkosť susedov určujeme silou signálu, teda v tomto prípade sused s najsilnejším signálom) z množiny susedov, ktoré majú rovnakú (minimálnu) vzdialenosť od sink-u v počte hopov ($\mathcal{N}_i^* = \{j \mid h_j =$

⁴Erupcia sopky bude zaznamenaná oveľa viac uzlami a bude korešpondovať s väčším k ako táborový oheň.

⁵Algoritmus vytvára len jeden strom, pričom optimálne stromy pre rôzne parametre súvislosti sa môžu líšiť.

⁶Celkový objem komprimovaných dát poslaných po linkách stromu.

$\min_{m \in \mathcal{N}_i} h_m, j \in \mathcal{N}_i\}$). MinDist vychádza z toho, že najviac sa budú prekrývať oblasti snímania najbližších susedov, pričom poradie vyberáme tak, aby sme postupovali smerom k sink-u. Nech i je uzol stromu a $p(i)$ označuje jeho otca. Nech množina $\{j_1, j_2, \dots, j_n\}$ je množina susedov i vo vzdialenostiach $d_{i,j_1} \leq d_{i,j_2} \leq \dots \leq d_{i,j_n}$, a teda sila signálov prijímaných od susedných uzlov je $e_{i,j_1} \geq e_{i,j_2} \geq \dots \geq e_{i,j_n}$. Potom uzol i má otca

$$p(i) = \{j \mid e_{i,j} = \max_{k \in \mathcal{N}_i^*} e_{i,k}, j \in \mathcal{N}_i^*\}$$

- *Maximálna vzdialenosť k ďalšiemu hopu (MaxDist)* - je opakom MinDist. Snaží sa o čo najväčší krok po ceste smerom k sink-u, ale na úkor možnosti pre lepšie odstraňovanie zbytočných dát.

$$p(i) = \{j \mid e_{i,j} = \min_{k \in \mathcal{N}_i^*} e_{i,k}, j \in \mathcal{N}_i^*\}$$

- *Náhodný ďalší hop (Random)* - algoritmus volí uzly náhodne s uniformnou pravdepodobnosťou spomedzi kandidátov s minimálnym počtom hopov k sink-u. Jeho výkonnosť je niekde medzi MinDist a MaxDist. Nech $uniform(\mathcal{A})$ je funkcia, ktorá náhodne uniformne zvolí prvok z množiny \mathcal{A} . Potom otec uzla i v agregáčnom strome pre Random heuristiku je

$$p(i) = \{j \mid j = uniform(\mathcal{N}_i^*)\}$$

- *Maximálny stupeň ďalšieho hopu (MaxDegree)* - zvolí uzol spomedzi susedných uzlov s minimálnym počtom hopov k sink-u, ktorý má najväčšiu kardinalitu množiny susedov. Takýto uzol má pravdepodobne veľa prekrývajúcich sa oblastí snímania so svojimi susedmi a teda jeho zvolením môžeme odstrániť značné množstvo prebytočných dát u veľkého počtu senzorov tak skoro ako sa len dá.

$$p(i) = \{j \mid |\mathcal{N}_j| = \max_{k \in \mathcal{N}_i^*} |\mathcal{N}_k|, j \in \mathcal{N}_i^*\}$$

- *Maximálna podmnožina ďalšieho hopu (MaxSubset)* - taktiež vyberá uzly podľa množiny susedov susedných uzlov. Nech j je sused uzla i . Môže nastať situácia, že niektoré uzly zo susedstva j si ho nezvolia ako svojho rodiča, pretože majú menší počet hopov k sink-u. MaxDegree medzi takýmito uzlami nerozlišuje, a tým nemusí zvoliť najlepšiu možnosť. Preto MaxSubset vyberá uzly podľa *čiasťného stupňa*, ktorým je počet uzlov v komunikačnom dosahu j , ktorých minimálny počet hopov k sink-u je väčší ako u uzla j ($\mathcal{N}'_j = \{t \mid h_t > h_j, t \in \mathcal{N}_j\}$). Potom

$$p(i) = \{j \mid |\mathcal{N}'_j| = \max_{k \in \mathcal{N}_i^*} |\mathcal{N}'_k|, j \in \mathcal{N}_i^*\}$$

Výhodou takýchto heuristik je, že používajú len lokálne informácie a nemusia poznať svoju polohu v sieti. Naopak pre zistenie jednotlivých množín susedov je potrebné poslať isté množstvo správ, ktoré pri veľkých a hustých sieťach môžu mať značný dopad na energetickú spotrebu.

5.2 Agregácia vytváraním súhrnov dát

Senzorové siete sú náchylné na časté výpadky, poruchy uzlov. Pri agregácii pomocou stromu práve tieto zlyhania uzlov alebo komunikácie predstavujú najväčšie riziko. Zlyhanie spôsobuje stratu celého podstromu meraní v danom uzle, a tak sa veľa informácií k sink-u nedostane. Bežné výpadky v sensorovej sieti tak môžu spôsobiť výrazné chyby v odpovedi na požiadavku, ktorá bola na sieť kladená [43]. Prirodzeným riešením je preposielanie strategických správ, čím sa však výrazne zvyšujú energetické výdavky siete a znižuje jej čas odozvy, čo je pri sensorových sieťach neželaným javom. Riešením by bolo aj použitie iných štruktúr ako sú stromy, pri ktorých by sa dáta posielali viacerým rodičom, čo však tiež nevedie k úplnému odstráneniu daného problému.

Je zrejmé, že agregácia pomocou stromov je úzko previazaná s routovaním v sensorovej sieti. Snaha o odstránenie straty údajov potom vedie k routovaniu cez viacero ciest v sieti⁷, čo vedie k duplicite správ, nevyhovujúcemu stavu pri agregácii. Napríklad ak sa nejaké meranie pošle po štyroch rôznych cestách (pre zvýšenie pravdepodobnosti, že aspoň jedna správa dorazí do cieľa), môže nastať situácia, kedy táto informácia bude započítaná až trikrát a nie raz.

S. Nath a kol. [43] predstavili *šírenie súhrnov*⁸, schému pre kombinovanie viac-cestných routovacích protokolov s algoritmami pre vyhýbanie sa duplicite správ. Oddelením agregácie od routovania tak pri šírení súhrnov môže byť použitá akákoľvek viac-cestná routovacia schéma. Oddelenie sa dosahuje cez súhrny nezávislé od poradia a duplicity (ODI súhrny)⁹. Ide o prehľady s malou veľkosťou, obsahujúce čiastočné výsledky, ktoré boli uzlom prijaté, pričom každé konkrétne meranie je v nich zarátané len raz. Nezávisia teda od poradia v akom boli merania do uzla doručené, ani od ich počtu. Pri agregácii sa na súhrn aplikujú nasledovné funkcie:

- *generovanie súhrnu (Synopsis Generation - SG())* - vezme merania senzorov a vygeneruje súhrn reprezentujúci tieto dáta,

⁷multi-path routing

⁸synapsis diffusion

⁹order- and duplicate-insensitive synopses

- *zlúčenie súhrnov (Synopsis Fusion - SF())* - z dvoch súhrnov vytvorí jeden nový,
- *zhodnotenie súhrnu (Synopsis Evaluation - SE())* - transformuje súhrn do konečného výsledku.

Schéma šírenia súhrnov pozostáva z dvoch fáz. Najprv sa v *distribučnej fáze* dotaz šíri sieťou takzvanou technikou zaplavovania¹⁰ a vytvorí sa agregáčna topológia. V *agregačnej fáze* sa agregované hodnoty routujú postupne späť k dotazujúcemu uzlu. Počas tejto fázy uzly opakovane používajú SG() a SF() funkcie. Uzol sníma dáta, z ktorých pomocou SG() funkcie vytvára súhrny, pričom po obdržaní súhrnu od iného uzla použije SF() funkciu na aktualizovanie svojho lokálneho súhrnu. Nakoniec uzol, ktorý zadal dotaz, použije SE() funkciu na získanie výsledku.

Ďalším dobrým príkladom agregácie pomocou vytvárania súhrnu dát je sumarizačná štruktúra *q-digest*¹¹. Ide o štruktúru s pevne danou, užívateľom určenou veľkosťou, ktorá aproximatívne určuje distribúciu nasnímaných dát [44]. Q-digest má zaujímavé vlastnosti:

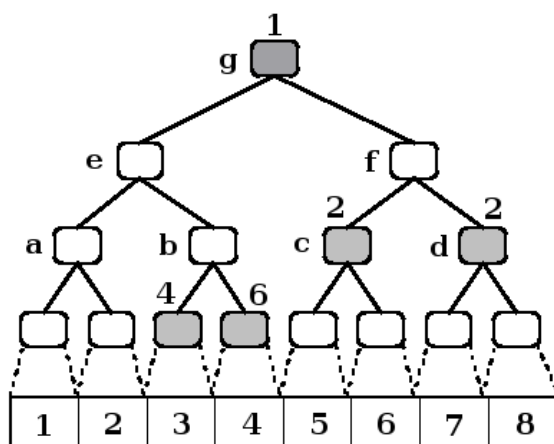
- *Vyváženie chyby a pamäte* - užívateľ si sám zvolí veľkosť q-digestov, ktorú bude sieť používať. Pre aplikácie citlivé na presné dáta sa zvolí väčšia veľkosť správ, naopak pri snahe o šetrenie energie sa zvolí menšia veľkosť, pričom q-digest bude pre zvolené hranice ponúkať čo najlepšie garancie chýb.
- *Faktor spokojnosti* - q-digest v sebe obsahuje aj mieru najväčšej naakumulovanej chyby, teda spolu s výsledkom máme aj pravdepodobnosť jeho chybovosti.
- *Viacnásobné dotazovanie* - po vykonaní q-digest dotazu sa v sink-u (alebo základnej stanici¹²) nachádzajúci q-digest obsahuje informácie, z ktorých môžeme získať aj množstvá a distribúciu dát alebo zhody údajov bez ďalšieho dotazovania siete.

Každý uzol má vlastný q-digest, ktorý reprezentuje dáta dostupné uzlu. Q-digest pozostáva z množiny rôzne veľkých priehradok a im prislúchajúcim počítadlom. Množina priehradok prichádzajúcich do úvahy sa volí z binárneho rozdelenia priestoru hodnôt $1 \dots \sigma$ reprezentovaného binárnym stromom T výšky $\log \sigma$. Každý vrchol stromu $v \in T$ môže predstavovať priehradku

¹⁰flooding - technika, pri ktorej uzol po prijatí správy prepošle správu všetkým svojim susedom

¹¹quantile digest

¹²base station



Obrázok 5.2: Kompletný binárny strom pre q-digest

Strom T je vybudovaný nad celým rozsahom $[1 \dots \sigma]$ hodnôt dát. Listy predstavujú jednoduché hodnoty. Tmavé uzly sú zahrnuté v q-digest-e Q a čísla nad nimi reprezentujú ich počty.

s rozsahom $[v : \min, v : \max]$. Koreň má rozsah $[1, \sigma]$ a synov s rozsahmi $[1, \frac{\sigma}{2}]$ a $[\frac{\sigma}{2} + 1, \sigma]$. Listy majú priehradky o veľkosti 1, teda samotné hodnoty (obrázok 5.2, podľa [44]).

V senzorum uzle sa nachádza q-digest Q ako podmnožina možných priehradok zo stromu T s im prislúchajúcimi počítadlami ($count(v)$). Počet hodnôt nameraných medzi 1 a $\frac{\sigma}{2}$ je počet všetkých uzlov v podstrome s koreňom v uzle $[1, \frac{\sigma}{2}]$. Na obrázku 5.2 uzol f korešponduje s rozsahom $[5 \dots 8]$ a počet všetkých hodnôt v tomto rozsahu je $2 + 2 = 4$. Veľkosť q-digest-u určuje parameter kompresie k . Pri danom parametri k sa uzol v nachádza v q-digest-e vtedy a len vtedy ak spĺňa nasledujúce podmienky:

$$count(v) \leq \left\lfloor \frac{n}{k} \right\rfloor$$

$$count(v) + count(v_p) + count(v_s) > \left\lfloor \frac{n}{k} \right\rfloor$$

kde v_p je otcem a v_s súrodencom uzla v . Tieto podmienky neplatia pre koreň (nemá otca ani súrodenca, teda nemusí spĺňať druhú podmienku) a pre listy. List patrí do q-digest-u ak jeho počítadlo je väčšie ako $\lfloor \frac{n}{k} \rfloor$.

Takýmto spôsobom sú tvorené prehľady o výskytoch nameraných hodnôt v sieti. Čím väčšiu veľkosť užívateľ pre q-digest povolí, tým užšie rozsahy pre namerané hodnoty získa a teda chyba bude menšia.

Agregácia pomocou stromov a prehľadov predstavuje najviac využívané techniky pre agregáciu dát. Samozrejme existujú aj iné prístupy, viac alebo menej odvodené od týchto dvoch. Tvorenie stromu môžeme zameniť za problém prietoku siete pri obmedzenej kapacite uzlov (uzlom u môže pretiecť najviac w_u informácií), ktorého riešenie sa však nedá aplikovať priamo z teórie grafov kvôli decentralizácii a potrebe adaptívnosti algoritmu [45]. Na techniku prehľadov môžeme aplikovať myšlienku ich uloženia a postupného progresívneho starnutia [46]. Pre požiadavky špecifických aplikácií je nutné sa zamyslieť aj nad bezpečnosťou pri agregovaní dát [47] alebo ich kvalitou pri chýbajúcich informáciách a hluku prostredia [48]. Prvoradou úlohou však stále ostáva znížiť objem prenášaných dát sieťou čo najjednoduchším (minimálne režijné náklady) spôsobom pre minimalizovanie energetickej spotreby.

Kapitola 6

Routovanie

Vhodný routovací algoritmus je dôležitý pre každú sieť. Závisia od neho celkové vlastnosti siete a je kostrou jej fungovania. V senzorových sieťach je dôraz na použitie dobrej routovacej techniky ešte väčší. Ovplyvňuje koľko správ a po akých linkách bude sieťov prechádzať, a teda od jeho voľby závisí aj celkové množstvo komunikácie v sieti. Keďže v senzorových sieťach sa snažíme o čo najnižšiu spotrebu energie, potrebujeme minimalizovať cesty, po ktorých sú správy posielané. Okrem obmedzenia energetických zdrojov však senzorová sieť vyžaduje routovanie, ktoré sa prispôsobuje meniacim podmienkam kvôli chybovosti, výpadkom uzlov alebo ich mobilite, teda neustále sa meniacej topológii. Ďalej treba brať do úvahy veľkosť a hustotu ad hoc senzorových sietí a chýbajúcu centrálnu autoritu. Z vyššie spomínaných dôvodov je zrejmé, že existujúce routovacie algoritmy pre klasické alebo bezdrôtové siete sú pre senzorové siete nevhodné.

Ako u všetkých protokolov pre senzorové siete aj routovací bude zameraný na šetrenie energie uzlov, na čo sa môžeme pozerať z dvoch strán. Ako na energeticky efektívne algoritmy sledujúce ostávajúcu energiu v uzloch alebo ako na algoritmy, ktoré sledujú energiu potrebnú na prenos správ. Nasledujúce techniky sú založené na spomínaných atribútoch [11]:

- *Cesta s maximálnou zostávajúcou energiou* - volí sa cesta k sink-u, ktorá má maximálnu zostávajúcu energiu, teda súčet energetických zdrojov uzlov pozdĺž cesty. Zvolená cesta však nemusí byť najkratšou, ktorá vedie k sink-u a teda ani optimálnou.
- *Cesta s minimálnou energiou* - volí sa cesta medzi vysielajúcim uzlom a sink-om, ktorá spotrebuje minimálne množstvo energie na doručenie paketu.
- *Cesta s minimom hopov* - algoritmy založené na tejto technike vyberajú

cestu s minimálnym počtom hopov. Ak je energia potrebná na prenos paketu pre každú linku rovnaká, potom sú techniky pre minimálnu energiu a minimum hopov ekvivalentné.

- *Cesta s uzlom s maximálnou minimálnou zostávajúcou energiou* - preferuje sa cesta, na ktorej uzol s minimálnou zostávajúcou energiou má väčšie energetické zdroje ako uzly s minimálnou zostávajúcou energiou iných ciest.

Hoci sú tieto techniky založené na výbere cesty s požadovanou vlastnosťou, algoritmy sú distribuované a lokalizované. Každý uzol pozná topológiu len do vzdialenosti malého počtu hopov, väčšinou len svojich susedov. Preto sa každý uzol nezávisle rozhoduje, ktorému susedovi pošle paket.

Príkladom routovacieho algoritmu založenom na výbere cesty s minimálnou energiou môže byť *Power progress* algoritmus [49]. Nech uzol práve držiaci paket je S , cieľom je D a A je susedom S . Nech $|SA| = r$, $|SD| = d$ a $|AD| = x$, kde $x < d$. Energia potrebná na poslanie správy z S do A je $r^\alpha + c$ pre dané α a c , pričom sa dosiahne pokrok $(d - x)$. Ak by sme takto postupovali ďalej, mali by sme $\frac{d}{d-x}$ krokov s celkovou cenou $(r^\alpha + c)\frac{d}{d-x}$. Preto uzol S pošle paket tomu susedovi, ktorý minimalizuje $(r^\alpha + c)\frac{d}{d-x}$, teda minimalizuje sa energia spotrebovaná na jednotku pokroku (posunu k cieľu).

6.1 Geometrické routovanie

Routovanie aké poznáme z Internetu je založené na IP adresách. V senzorných sieťach však neexistuje univerzálny systém pre adresy. Rodina routovacích algoritmov, ktoré používajú geografické koordináty uzlov ako adresy pre účely routovania, sa nazýva *geometrické routovanie* [50]. Uzol si je vedomý svojej pozície a pozície svojich susedov. Ďalším predpokladom je, že zdroj správy pozná pozíciu cieľového uzla. Pokiaľ prvý predpoklad sa dá splniť preposlaním konštantného počtu správ medzi susednými uzlami, druhý už vyžaduje lokalizáciu uzlov¹. Existujú aj scenária, kde stačí správu poslať akémukoľvek uzlu v určitej oblasti, čo nazývame *geocasting*.

Najjednoduchším príkladom geometrického routovania je greedy routovanie. Pri tomto routovaní je správa v každom uzle zasielaná susedovi, ktorý je najbližšie k cieľu. Algoritmus sa však môže dostať do *mŕtveho bodu*, uzla, ktorého všetci susedia sú od cieľa vzdialený viac ako on sám. Tieto prípady sa však dajú riešiť face routingom, ktorý prechádza hranice oblastí v planárnom

¹Uzly poznajú svoju geografickú polohu. Túto problematiku rozoberáme v ďalších kapitolách.

grafe siete. Kombináciou týchto dvoch routovacích techník dostávame jednoduchý a efektívny routovací algoritmus, ktorý garantuje doručenie správ, *GOAFR⁺* [50]. Algoritmus používa parametre ρ_0 , ρ a σ , ktoré sú počas výpočtu konštantné, ale musia spĺňať podmienky $1 \leq \rho_0 < \rho$ a $0 < \sigma$. Tieto parametre sa zvolia pred vykonávaním algoritmu a môžu tak prispôbiť routovanie vlastnostiam siete². *GOAFR⁺* pracuje nasledovne:

- Začína v zdrojovom uzle s ³. Inicializuje kružnicu C so stredom v t a polomerom $r_C := \rho_0 |\overline{st}|$.
- *Greedy routovanie* - opakuje kroky greedy algoritmu pokiaľ nedosiahne uzol t , kedy algoritmus skončí, alebo lokálne minimum. Ak dosiahne lokálne minimum, prepne sa na face routovanie. Pri každom navštívenom uzle zmenší polomer kružnice C na $r_C = \frac{r_C}{\rho}$ pod podmienkou, že uzol sa ešte nachádza vnútri kružnice.
- *Face routovanie* - nech u_i je práve navštívené lokálne minimum. Začne preskúmať hranicu oblasti grafu F_i , ktorá obsahuje spojnicu $\overline{u_i t}$. Po preskúmaní F_i a návrate do u_i sa presunie do uzla, ktorý je najbližšie k t a už bol navštívený a prepne sa do greedy routovania. Ak po preskúmaní F_i žiaden z navštívených uzlov nie je bližšie k t ako u_i , graf je nespojitý a neexistuje v ňom cesta vedúca z vrcholu s do vrcholu t . Počas prehľadávania hranice oblasti F_i sa aktualizujú počítadlá p pre počet uzlov F_i , ktoré sú bližšie k t ako u_i a q pre počet uzlov z F_i , ktoré sa nachádzajú ďalej od t ako u_i . Pritom vykonáva nasledovné špeciálne akcie:
 1. ak pretne kružnicu C po prvý raz, otočí sa a bude preskúmať hranicu F_i v opačnom smere,
 2. pri druhom pretnutí kružnice C , ak žiaden z doposiaľ navštívených uzlov nie je bližšie k t ako u_i , zväčší C ($r_C := \rho r_C$) a pokračuje vo face routingu, akoby začínal v u_i . Inak sa presunie do uzla, ktorý je najbližšie k t a pokračuje v greedy routovaní,
 3. ak $p > \sigma q$, teda už bolo navštívených o konštantný faktor viac uzlov z F_i bližšie k t než u_i ako tých vzdialenejších, presunie sa do zatiaľ najbližšieho uzla k t a pokračuje v greedy routovaní.

²Kuhn a kol. podľa simulácií uvádzajú $\rho_0 = 1.4$, $\rho = \sqrt{2}$ a $\sigma = \frac{1}{100}$ ako praktické hodnoty.

³Zdrojový uzol pozná polohu cieľového uzla, ostatné uzly poznajú polohy svojich susedov.

Napriek svojej jednoduchosti tento algoritmus dosahuje dobré výsledky a aj v najhoršom prípade je asymptoticky optimálny [50].

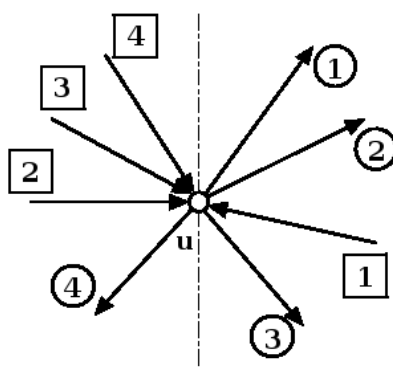
Geometrické routovanie komplikujú dva faktory [51]. Prvým je lokalizácia. Cena a energetická náročnosť GPS vedie k použitiu polohovacích algoritmov, ktoré zasa zaťažujú sieť výpočtovo aj komunikačne. Druhým faktorom je, že presné koordináty uzlov môžu byť nevýhodou, pretože nezahŕňajú prekážky a iné topologické vlastnosti siete. Pre modelovanie týchto vzťahov existuje schéma, v ktorej si uzly najprv zvolia umelé virtuálne koordináty, a potom je na ne aplikovaný greedy algoritmus [52]. Takýto greedy algoritmus je potom oveľa spoľahlivejší. Papadimitriou a kol. [51] vyslovili zaujímavú domnienku, že každý planárny 3-súvislý graf obsahuje greedy vnorenie⁴. Greedy vnorenie je graf, v ktorom pre ľubovoľné dva vrcholy t a s existuje sused vrchola s , ktorého vzdialenosť od t je menšia ako má s . Teda každý planárny 3-súvislý graf sa dá vložiť do roviny takým spôsobom, že medzi ľubovoľnými dvoma vrcholmi s a t existuje v takomto grafe vzdialenosť znižujúca cesta. Keďže vkladanie hrán nenarušuje túto vlastnosť, pre každý graf, ktorý obsahuje ako podgraf planárny 3-súvislý graf, existujú súradnice, pre ktoré je greedy routovanie vždy úspešné.

6.2 Geometrické routovanie v orientovaných grafoch

Predchádzajúce routovacie algoritmy boli založené na predpoklade, že komunikačné linky medzi uzlami sú obojsmerné, teda že komunikačné dosahy jednotlivých uzlov sú rovnaké a nemenia sa. Táto vlastnosť sa však v praxi nedá vždy dodržať. Môže to spôsobovať viacero faktorov. Prírodné prekážky znemožňujú priamy kontakt medzi uzlami, alebo znižujú silu signálu. Uzly môžu mať aj rôzne komunikačné dosahy. Routovacie algoritmy v takýchto prípadoch modelujú sieť pomocou orientovaných grafov. Routovanie je sťažené aj faktom, že uzol má len lokálne informácia, čo môže viesť k vzniku slučiek [54].

Ak by sme vzali do úvahy neorientovaný planárny graf, postačuje nám face routing. Vnútorne oblasti môžeme prechádzať v protismere hodinových ručičiek a vonkajšie v smere hodinových ručičiek. Ide o známe pravidlo pravej ruky. Ak by sme v bludisku položili pravú ruku na stenu a šli dopredu, navštívime všetky steny. Ako rutinu používa túto techniku compass routovanie [55]. V orientovanom grafe však môžeme pri tomto prechádzaní natrafiť na protismernú hranu.

⁴greedy embedding



Obrázok 6.1: Očíslovanie hrán

Čísla v kruhoch reprezentujú usporiadanie odchádzajúcich hrán a čísla vo štvorcoch prichádzajúcich hrán do uzla v .

Tento problém riešia Chávez a kol. v [54] tým, že pri natrafení na protismernú hranu nájdú cestu v grafe na jej druhý koniec. Predpokladom je eulerovský planárny graf, teda graf, v ktorom každý uzol má rovnaký počet prichádzajúcich a odchádzajúcich hrán. Prichádzajúce hrany sú v každom uzle očíslované v smere hodinových ručičiek začínajúc od dvanástej hodiny, podobne aj odchádzajúce (obrázok 6.1, podľa [54]). Na tomto očíslovaní je definovaná funkcia $succ(e)$, ktorá pre i -tú prichádzajúcu hranu e do uzla vráti i -tú odchádzajúcu hranu z uzla. Opakovaným aplikovaním funkcie $succ(e)$ získame uzavretý ťah začínajúci a končiaci v tej istej hrane e . Routovanie v orientovanom grafe potom prebieha nasledovne:

- prebieha rutina pre traverzovanie⁵ oblasti grafu,
- vždy keď je potrebné prejsť po protismernej hrane e , traverzujú sa nasledovné hrany:

$$succ(e), succ(e)^2, succ(e)^3, \dots, succ(e)^k$$

kde $succ(e)^{k+1} = e$. Po prejdení hranou $succ(e)^k$ sa opäť používa pôvodné traverzovanie oblasti.

Routovanie v orientovaných grafoch sa javí ako zložitý problém a okrem zaplavovania⁶ sa zdá, že neexistuje jednoduché rozšírenie routovacích algoritmov pre neorientované grafy, ktoré by sa dalo aplikovať na orientované.

⁵Technika pre prechádzanie grafom, pri ktorej sú navštívené všetky uzly.

⁶flooding

Chávez a kol. taktiež prezentujú sofistikovaný routovací algoritmus pre silne súvislé⁷ zvonku planárne geometrické siete⁸. Planárna geometrická sieť G je zvonku planárna, ak jeden z komponentov F obsahuje všetky uzly (vonkajšia oblasť). Predpokladom je, že táto oblasť tvorí konvexný polygón v \mathbb{E}^2 . Za silne súvislú považujeme geometrickú sieť vtedy, ak pri fixnej orientácii pre každú usporiadanú dvojicu vrcholov existuje (priama) cesta, ktorá ich spája. Orientácia v geometrickej sieti G :

- pre danú trojicu vrcholov x, y a z je $V_{\curvearrowright}(x, y, z)$ (respektíve $V_{\curvearrowleft}(x, y, z)$) usporiadaná množina vrcholov rôznych od x a z , ktoré sú navštívené pri prechádzaní vonkajšej oblasti G z uzla y v protismere (respektíve v smere) hodinových ručičiek pokiaľ sa nedosiahne uzol x alebo z ,
- nech $N_{\curvearrowright}(x, y, z) = V_{\curvearrowright}(x, y, z) \cap N^+(x)$ kde $N^+(x)$ je množina hrán odchádzajúcich z vrcholu x a nech $N_{\curvearrowleft}(x, y, z) = V_{\curvearrowleft}(x, y, z) \cap N^+(x)$,
- ak $N_{\curvearrowright}(x, y, z) \neq 0$ potom $v_{\curvearrowright}(x, y, z)$ je prvý vrchol z $N_{\curvearrowright}(x, y, z)$. Podobne definujeme $v_{\curvearrowleft}(x, y, z)$ ako prvý uzol z $N_{\curvearrowleft}(x, y, z)$, ak existuje.

Týmto spôsobom určíme fixnú orientáciu pre traverzovanie v geometrickej sieti. Ako vidno routovanie v orientovaných sieťach nie je triviálny problém a existujúce routovacie algoritmy potrebujú pre svoje správne fungovanie silné predpoklady.

Keďže sensorové siete môžu byť mobilné a sú náchylné na zlyhania uzlov alebo komunikácie, je vytvorenie routovacích ciest a hlavne ich udržiavanie problémom. Často sa rieši *dočasným stavom* u routovacích protokolov. Časovače pri routovacích záznamoch sa obnovujú pri prijatí paketu alebo nastáva periodická aktualizácia. Ak sa komunikačná linka preruší, časovaču nakoniec vyprší čas, čo môže spustiť nápravu. Takýto prístup je výhodný v tom, že nepotrebuje kontrolné správy, a tým znižuje náklady na prevádzku protokolu. Nastavenie časovačov má nezanedbateľný vplyv na efektívnosť routovania [53].

6.3 Štruktúry vhodné pre geometrické routovanie

Keďže efektívnosť geometrického routovania ovplyvňuje celkový dizajn siete, do povedomia sa dostala otázka ako dizajnovať sensorovú sieť, aby podporovala efektívne routovacie techniky. Pre zjednodušenie môžeme predpokladať,

⁷strongly connected

⁸outerplanar

že sieť tvorí množina vrcholov v dvojrozsmernej rovine. Predpokladajme ďalej, že všetky uzly majú maximálny komunikačný dosah rovný nejakej jednotke. Potom sensorové uzly vytvárajú jednotkový kruhový graf, v ktorom je hrana medzi dvoma uzlami práve vtedy a len vtedy, ak ich euklidovská vzdialenosť je najviac jedna.

Problémom je teda zostrojiť podgraf jednotkového kruhového grafu, ktorý je riedky, môže byť skonštruovaný lokálne efektívnym spôsobom a stále je relatívne dobrý pre routovacie techniky v porovnaní s pôvodným jednotkovým kruhovým grafom [56]. Jednou z dôležitých vlastností takéhoto podgrafu by malo byť, aby najkratšia cesta medzi ľubovoľnými dvoma uzlami v podgrafe nebola oveľa väčšia ako je najkratšia cesta medzi týmito uzlami v pôvodnom jednotkovom kruhovom grafe. Túto vlastnosť zachytáva *predlžovací faktor* podgrafu. Podgraf, ktorý obsahuje všetky vrcholy pôvodného grafu, nazývame *spanner*. Pre potreby routovania bude našou snahou vytvoriť spanner s konštantným predlžovacím faktorom. Spanner, ktorý má len lineárny počet liniek, nazývame riedky spanner. Kvôli routovaniu v sensorových sieťach a vytvoreniu vhodnej virtuálnej chrbtice siete bolo študovaných niekoľko geometrických štruktúr. Nech $G = (V, E)$ je geometrický graf definovaný na V :

- *minimálna kostra grafu* ($MST(G)$)⁹ - ide o strom, ktorý je podgrafom G a spája všetky vrcholy z V pri minimálnej celkovej dĺžke hrán. Ide zjavne o najriedkejší možný podgraf, jeho predlžovací faktor môže byť až $n - 1$.
- *graf relatívneho susedstva* ($RNG(G)$)¹⁰ [57] - obsahuje len také hrany $(u, v) \in E$, pre ktoré neexistuje vrchol w taký, že hrany uw a wv v E spĺňajú $\|uw\| < \|uv\|$ a $\|wv\| < \|uv\|$. Teda hrana uv patrí do grafu ak prienik dvoch kruhov so stredmi v u a v neobsahuje žiadny vrchol z V taký, že hrany uw a wv sú v E . Graf relatívneho susedstva má predlžovací faktor $n - 1$.
- *Gabrielov graf* ($GG(G)$) [58] - graf obsahuje hranu uv z G vtedy a len vtedy, ak $disk(u, v)$ (kruh s priemerom uv) neobsahuje žiadny vrchol $w \in V$ taký, že $\|uw\| < \|uv\|$ a $\|wv\| < \|uv\|$. Gabrielov graf je planárny ak G je kompletý. Z definície takisto vidno, že $RNG(G)$ je podgrafom $GG(G)$ a pre neorientovaný súvislý graf G , $RNG(G)$ a $GG(G)$ sú súvislé a obsahujú minimálnu kostru grafu G . Predlžovací faktor u Gabrielovho grafu je medzi $\frac{\sqrt{n}}{2}$ a $\frac{4\pi\sqrt{2n-4}}{3}$ [59].

⁹minimum spanning tree

¹⁰relative neighborhood graph

- *Yaov graf* $(\overrightarrow{YG}_k(G))$ [56] - s parametrom $k \geq 6$ je definovaný: pre každý uzol u , každých k rovnako odedelených lúčov vznikajúcich v u definuje k kužeľov. V každom kuželi je zvolená najkratšia hrana uv zo všetkých hrán pri vrchole u , ak existuje a do podgrafu je pridaná orientovaná hrana \overrightarrow{uv} . Výsledný orientovaný graf sa nazýva Yaov graf. Ak by sme namiesto \overrightarrow{uv} pridávali hranu \overleftarrow{uv} , získali by sme reverz Yaovho grafu $\overleftarrow{YG}_k(G)$. Yaov graf má predĺžovací faktor $\frac{1}{1-2\sin\frac{\pi}{k}}$
- *Delaunayho triangulácia* $(Del(V))$ ¹¹ - predpokladajme, že neexistujú štyri vrcholy z V , ktoré by boli kocirkulárne. Trianguláciu V nazývame Delaunayho triangulácia, ak opísaný kruh každého z jej trojuholníkov neobsahuje vo svojom vnútri iné vrcholy z V ako sú tie, ktoré tvoria jeho hrany. Spĺňa takzvanú vlastnosť *prázdneho kruhu* - pre každú hranu existuje kruh obsahujúci koncové body hrany, ktorý neobsahuje žiadne iné body z V . Obdobne trojuholník nazývame *Delaunayho trojuholník*, ak jeho opísaný kruh neobsahuje ostatné body z V . *Voronoiho oblasť*¹² $Vor(p)$, vrcholu $p \in V$, je súbor bodov s dvojrozmernými súradnicami, ktoré sú bližšie k p ako k akémukoľvek inému vrcholu z V . *Voronoiho diagram* pre V je zjednotením všetkých Voronoiho oblastí $V(p)$, kde $p \in V$. Delaunayho triangulácia je duálna k Voronoiho diagramu. Dva vrcholy p a q sú spojené v $Del(v)$ práve vtedy a len vtedy, ak $Vor(p)$ a $Vor(q)$ majú spoločnú hranicu. Spoločná hranica dvoch Voronoiho oblastí $Vor(p)$ a $Vor(q)$ sa nachádza na kolmici sečnice segmentu pq . Segment hranice Voronoiho oblasti nazývame *Voronoiho hrana* a priesečník dvoch Voronoiho hrán nazývame *Voronoiho vrchol*. Voronoiho vrchol tak tvorí stred kruhu, ktorý opisuje nejaký Delaunayho trojuholník.

Dôležitým faktom je, aby sa dali tieto štruktúry skonštruovať lokálne. To znamená, že každý uzol vie určiť hrany, ktoré s ním budú incidentné, len z lokálnej informácie o susedoch vzdialených najviac konštantný počet hopov. Z definície vidno, že Yaov graf, graf relatívneho susedstva a Gabrielov graf sa dajú zostrojiť lokálne. Bohužiaľ euklidovská minimálna kostra a Delaunayho triangulácia sa nedá zostrojiť žiadnym lokálnym algoritmom [56]. Gabrielov graf sa používa v niektorých face routovacích protokoloch [60], graf relatívneho susedstva sa zasa používa pre efektívne broadcastovanie v one-to-one broadcastovacou modeli [61].

Nech teraz $d_G(u, v)$ je celková cena najkratšej cesty medzi vrcholmi u a v v hranovo ohodnotenom grafe G . Pre danú množinu bodov v dvojrozmernej

¹¹Delaunay triangulation

¹²Voronoi region

rovine, podgraf $G = (V, E)$ grafu H je t -spanner grafu H ak pre ľubovoľnú dvojicu vrcholov u a v platí, že $d_G(u, v) \leq t \cdot d_H(u, v)$. Predpokladajme, že cena hrany uv je rovná euklidovskej vzdialenosti medzi vrcholmi u a v . Ak graf H je kompletný, označujeme G iba ako t -spanner. G nazývame riedkym, ak obsahuje len $O(n)$ hrán, kde $n = |V|$. Navyše ak celková dĺžka hrán v G je iba konštantným faktorom euklidovskej minimálnej kostry V , potom G je lacný¹³. Je známe, že graf relatívneho susedstva ako aj Gabrielov graf nie sú spanner. Naopak, Delaunayho triangulácia je t -spanner, kde t je zhora ohraničené, $t \leq \frac{4\sqrt{3}}{9}\pi$ [62].

Ako príklad uvedieme jeden z algoritmov pre konštrukciu spannera. Ide o metódu predstavenú v [62], ktorá konštruuje lacný planárny spanner s ohraničeným stupňom uzlov z množiny bodov V v dvojrozmernej rovine. Predpokladá však existenciu unikátnych identifikátorov pre každý uzol u , označených $ID(u)$. Algoritmus pracuje nasledovne:

1. vypočíta Delaunayho trianguláciu množiny uzlov V , $Del(V)$. Nech $N_{Del}(u)$ sú susedia uzla u v Delaunayho triangulácii a nech d_u je stupeň uzla u v $Del(V)$. $N_{del}(u)$ a d_u sa dajú ľahko vypočítať v lineárnom čase, lebo Delaunayho triangulácia má len lineárny počet hrán.
2. nájde usporiadanie π pre V nasledovne: Nech $G_1 = Del(V)$ a $d_{G_1, u}$ je stupeň u v grafe G_1 . Odstráni uzol s najmenšou hodnotou ($d_{G_1, u}, ID(u)$) a všetky hrany s ním incidentné z grafu G_1 . Priradí $\pi_u = n - i + 1$ a pomenuje vzniknutý graf G_{i+1} . Opakuje túto procedúru pre $1 \leq i \leq n$. Nech P_v predstavuje predchodcov v v π , teda $P_v = \{u \in V : \pi_u < \pi_v\}$. Pretože G_i je planárny, vieme, že najmenšia hodnota $d_{G_i, u}$ je najviac 5. Potom pre usporiadanie π má uzol u najviac 5 hrán k svojim predchodcom P_u v $Del(V)$.
3. nech E je množina hrán z $Del(V)$ a nech E' je množina hrán pre požadovaný spanner. Algoritmus inicializuje E' ako prázdnu množinu a označí všetky vrcholy vo V ako *nespracovaný*. Potom pre každý uzol u vo V podľa vzostupného poradia π vykonáva nasledujúce kroky pre pridanie niektorých hrán z E do E' :
 - (a) nech v_1, v_2, \dots, v_k je označenie predchodcov uzla u . Keďže predchodcov môže byť najviac 5 (už spracovaných Delaunayho susedov), $k \leq 5$. Potom pri uzle u je $k \leq 5$ otvorených sektorov, ktorých hranice tvoria lúče vychádzajúce z u k spracovaným susedom v_i uzla u v $Del(V)$. Každý takýto sektor sa potom rozdelí

¹³low weighted

na minimálne množstvo otvorených kužeľov stupňa najviac α , kde $\alpha \leq \frac{\pi}{2}$ je parameter.

- (b) pre každý takýto kužeľ, nech s_1, s_2, \dots, s_m sú geometricky usporiadaní¹⁴ susedia $N_{Del}(u)$ uzla u v tomto kuželi. Teda s_1, s_2, \dots, s_m sú všetko nespracované uzly spojené nejakou hranou z E s uzlom u v tomto kuželi. Algoritmus najprv vyberie najkratšiu hranu z E spojenú v kuželi s u a pridá ju do E' . Potom pridá do E' všetky hrany $s_j s_{j+1}$, $1 \leq j < m$. Z definície Delaunayho triangulácie platí, že $s_j s_{j+1} \in E$.
- (c) označí uzol u ako spracovaný.

Táto procedúra sa opakuje pre vzostupné usporiadanie π , pokiaľ nie sú všetky uzly označené ako spracované.

4. ak je potrebná aj lacná štruktúra, na vzniknutý spanner sa aplikuje greedy spanner algoritmus od Gudmundssona a kol. [63].

Nevýhodou uvedeného algoritmu však je, že je centralizovaný. Na rozdiel od iných centralizovaných algoritmov pre výpočet spannera sa ale tento dá rozšíriť na lokálny, i keď rozšírenie nie je priamočiare.

¹⁴Usporiadanie v smere alebo v protismere hodinových ručičiek hrán us_1, us_2, \dots, us_m .

Kapitola 7

Broadcastovanie

Podľa princípu, na akom pracujú senzorové siete, ich môžeme rozdeliť do troch skupín. Na siete, ktoré permanentne snímajú dáta a posielajú ich k základným staniciam, ktoré sa nazývajú aj sink-y. Na siete, ktoré sú spúšťané nastaním určitej situácie (vznik požiaru, vniknutie narušiteľa) a nakoniec na siete, ktorých aktivitu snímania určuje užívateľ tak, že na ne kladie dotazy. Routovacie algoritmy sa v týchto sieťach používajú na doručenie nasnímaných dát k sink-u. V tretej skupine senzorových sietí je však potrebné najprv zadať sieti úlohu, ktorú má plniť, a to každému uzlu kvôli jej distributívnosti. Na tento účel sa používa broadcastovanie, väčšinou spojené s adresáciou hodnotami¹. Broadcastovanie je štandardným one-to-many prístupom, pri ktorom potrebujeme zaslať rovnakú správu viacerým uzlom, väčšinou uzlom v celej sieti.

7.1 Broadcastovanie založené na zaplavovaní

Obdobne ako pri routovaní, ani pri broadcastovaní nie je vhodné používať broadcastovacie algoritmy z pevných alebo bezdrôtových sietí kvôli obmedzeniam, ktoré majú senzorové ad hoc siete. Napriek tomu má zmysel uvažovať nad niektorými jednoduchými technikami ako je zaplavovanie², ktoré je použiteľné pri ľubovoľnej topológii a nezávislé od jej dynamickej zmeny. Je založené na jednoduchom princípe, kedy každý uzol po obdržaní správy (paketu) ho prepošle všetkým svojim susedom. V krajných prípadoch sa môže použiť aj na routovanie. Paket je vtedy preposielaný pokiaľ nedosiahne maximálny počet hopov, ktorý mu bol určený, alebo pokiaľ nedosiahne cieľový

¹Užívateľ nemá záujem o merania zo všetkých uzlov. Namiesto toho požaduje merania s hodnotami, ktoré spĺňajú jeho požiadavky.

²flooding

uzol. Zaplavovanie je síce implementačne a výpočtovo nenáročné, príliš však preťažuje sieť zbytočnou komunikáciou, čo je presný opak situácie, ktorú sa snažíme pri senzorových sieťach dosiahnuť. Jeho najväčšími nevýhodami sú [64]:

- *Zahlcovanie* - situácia, kedy sú tomu istému uzlu zasielané duplicitné správy. Napríklad ak uzol A má N susedov, ktorí sú taktiež susedmi uzla B , uzol B dostane N kópií správy zaslanej uzlom A .
- *Prekrývanie* - ak dva uzly snímajú rovnakú oblasť, môžu simultánne nasnímať rovnaké dáta. Susedia týchto uzlov potom dostanú duplicitné správy.
- *Ignorovanie zdrojov* - zaplavovanie neberie do úvahy energetické zdroje uzlov a je teda neefektívne v spotrebe energie.

Niektoré z problémov (najmä zahlcovanie) zaplavovania rieši jeho variácia, klebetenie³ [65]. Uzol po obdržaní správy ju nebroadcastuje všetkým susedom, ale zvolí si náhodne jedného z nich, ktorému pošle správu. Hoci sa výrazne zníži objem preposlaných správ, trvá dlhú dobu kým sa správa dostane do všetkých uzlov.

Iná technika, ktorá má zamedziť zbytočné preposielanie správ, je založená na predpoklade, že každý uzol pozná lokálnu topológiu do vzdialenosti dvoch hopov. Pri obdržaní správy potom uzol vyberie malú podmnožinu z uzlov, ktoré sú od neho vzdialené na jeden hop tak, aby pokryli (v zmysle rádiového dosahu) všetkých susedov vzdialených dva hopy. Podmnožinu 1-hop vzdialených susedov nazývame *preposielacia množina*⁴ [66] a problém jej nájdenia *problém minimálnej preposielacej množiny*⁵ [67]: nech A je zdroj a \mathcal{D} a \mathcal{P} sú množiny 1-hop a 2-hop vzdialených susedov uzla A . Problémom je nájdenie takej podmnožiny \mathcal{F} množiny \mathcal{D} , že každý uzol z množiny \mathcal{P} sa nachádza v komunikačnom dosahu aspoň jedného uzla z \mathcal{F} .

Ak neberieme do úvahy geografickú polohu uzlov, ide o známy *problém pokrytia množiny*⁶, ktorý sa dá riešiť Chvátalovým greedy algoritmom [68] s aproximačným faktorom $O(\log m)$, kde m je maximálna množina susedov. Algoritmus iteratívne vyberá uzol z 1-hop susedov, ktorý pokrýva najviac ešte nepokrytých 2-hop susedov a končí, keď sú všetci 2-hop susedia pokrytí.

Uvažujme teraz o tomto probléme v jednotkovom kruhovom grafe⁷. Obdobným problémom k problému minimálnej preposielacej množiny, je *prob-*

³gossiping

⁴forwarding set

⁵minimum forwarding set problem

⁶set cover problem

⁷unit-disk graph

lém *dominantnej množiny*⁸, ktorý je v jednotkových kruhových grafoch NP-ťažký [69]. Ďalším príbuzným problémom je *problém jednotkového kruhového pokrytia*⁹, ktorý je taktiež NP-ťažký [69]. Ani na jeden z nich sa však problém minimálnej preposielacej množiny nedá redukovať. Jeho zložitosť zatiaľ nie je známa, aj keď vieme, že je špeciálnym prípadom ďalšieho NP-ťažkého problému, *problému pokrytia kruhmi*¹⁰ [70]. Zrejme najvhodnejším kandidátom pre senzorové siete bude aproximačný algoritmus riešiaci problém minimálnej preposielacej množiny, ktorý je výpočtovo nenáročný. Štyri rôzne aproximačné algoritmy predstavujú Čălinescu a kol. v [67], najlepší s aproximačným faktorom 3.

7.2 Riadené šírenie

Jednou zo schém ako prebieha snímanie v senzorovej sieti je, že užívateľ zadá úlohu špecifickej oblasti, kde uzly začnú monitorovať prostredie. Po nasnímaní sú dáta routované späť k užívateľovi. Kvôli ušetreniu energie v sieti by bolo vhodné, aby sa routovacia cesta pre dáta počítala už počas broadcastovania úlohy. Na takomto princípe funguje riadené šírenie¹¹ [71]. Príklad jeho použitia môže vyzeráť takto: požiadavka ľudského operátora sa transformuje na záujem, ktorý sa šíri sieťou (broadcastuje) do určitého regiónu. Uzly po obdržaní záujmu začnú snímať svoje okolie pre požadovanú udalosť/dáta. Po jej nastaní sa dáta vracajú po opačnej ceste akou sa šírili záujem. Uzly po ceste môžu navyše prijaté dáta agregovať alebo ich aktualizovať vlastnými nasnímanými dátami.

Riadené šírenie pozostáva z nasledujúcich elementov: záujmy, dátové správy, gradienty a posilňovanie. Záujem je dotaz, ktorý špecifikuje, čo požaduje užívateľ od siete. Popisuje úlohu, ktorá je podporovaná senzorovou sieťou. Užívateľ zvolí dáta alebo udalosť, ktorú chce pomocou senzorovej siete monitorovať. Dáta sú pomenované dvojicou atribút-hodnota a môžu mať aj špecifikovanú oblasť, z ktorej sú požadované. Záujem sa do siete začne broadcastovať špeciálnym uzlom sink. Počas šírenia záujmu sa zároveň nastavujú gradienty v jednotlivých uzloch. Každý uzol, ktorý obdrží záujem, si nastaví smerník na suseda, od ktorého ho prijal. Týmto spôsobom sa vytvorí niekoľko ciest späť k sink-u.

⁸dominating set problem - požaduje podmnožinu uzlov dominujúcu (hranovo spojenú) všetkým ostatným uzlom

⁹unit-disk cover problem - minimálny počet jednotkových kruhov pokrývajúcich body v rovine

¹⁰disk cover problem - minimálna podmnožina daných kruhov, ktorá pokrýva dané body v rovine

¹¹directed diffusion

Po obdržaní záujmu v požadovanej oblasti začnú uzly snímať dáta. Po ich nasnímaní alebo po vyskytnutí udalosti popísanej záujmom, ich začínajú posielat' späť po viacerých cestách pomocou gradientov. Spočiatku sú posielané s malou frekvenciou, aby uvedomili sink, že nastala požadovaná udalosť. Sink do siete neustále posiela záujmy o dáta alebo udalosti a dostáva späť odpovede. V momente, keď sa do sink-u začínajú dostávať dáta, posilní sa jedna alebo menší počet ciest, ktoré budú preferované. Sink posilní jeden zo svojich susedov, aby začal dostávať reálne dáta (informácie s vyššou frekvenciou, teda presnejšie údaje o prebiehajúcej udalosti). Urobí to posielaním pôvodného záujmu v menších intervaloch (teda s vyššou frekvenciou). Uzol dostane záujem od suseda, na ktorého už má nastavený gradient a zistí, že záujem požaduje dáta s vyššou frekvenciou. Ak požiadavka na frekvenciu posielania dát je vyššia ako má akýkoľvek existujúci gradient v uzle, je uzol nútený posilniť jedného zo svojich susedov. Týmto spôsobom sa vytvorí cesta pre routovanie dát zo skúmanej oblasti smerom k sinku. Navyše sa podľa frekvencie prichádzajúcich dát spĺňajúcich požiadavky záujmu dynamicky mení aj routovacia cesta, teda sink dostáva aktuálne informácie aj pri pohybe sledovanej udalosti v sieti.

7.3 Šírenie informácie v mobilných ad hoc sieťach

Algoritmy v senzorových sieťach sú distributívne a uzly poznajú len svojich najbližších susedov. Niektoré broadcastovacie techniky na rozdiel od routovacích nepotrebujú poznať ani lokálnu geografickú polohu uzlov, čo umožňuje ich použitie aj v mobilných sieťach. Ako sa ale bude šíriť broadcastovaná informácia v takejto sieti?

Predstavme si k guľôčok, 1 červenú a $k - 1$ bielych, ktoré sa naháňajú po hranách grafu. Ak červená chyť niektorú z bielych, prefarbí ju na červenú a tá zas môže prefarbovať ostatné biele guľôčky. Kedy nastane stav, že všetky guľôčky sú červené? Tento jednoduchý prípad je inštanciou šírenia informácie náhodnými prechodmi grafu a modeluje šírenie informácie mobilnými uzlami [72].

Formálnejšie ide o model interagujúcich častíc, ktorých pohyb sa modeluje náhodným prechodom grafu. Jednotlivé prechody prebiehajú súčasne a pre zjednodušenie má počiatočná pozícia častíc rovnomernú distribúciu pravdepodobnosti prechodov. Nech k je počet častíc a $G(V, E)$ je graf prechodov, $|V| = n \geq k$, $|E| = m$. V počiatočnej fáze je jedna častica červená a ostatné biele. Každá z častíc prechádza grafom a pri interakcii s ostatnými sa sprá-

va podľa nasledovného pravidla: ak červená častica stretne (v jednom uzle grafu) jednu alebo viac bielych častíc, biele častice sa zmenia na červené.

Dimitriou a kol. v [72] používajú tento model na zistenie času prenášania správ medzi sensorovými uzlami v mobilnej sieti a navrhujú základné protokoly pre komunikáciu v takýchto sieťach [73]. Sú založené na náhodných stretnutiach mobilných uzlov a protokolom koordinovanom pohybe malej časti siete. Trojrozmernú oblasť reprezentujú *pohyblivým grafom*¹², ktorého uzly modelujú kockami s objemom podobným guli tvorenej komunikačným dosahom uzlov, a ktorého hrany sú medzi priľahlými kockami. Protokol funguje na nasledovnom princípe: podporné uzly sa pohybujú rýchlo, aby prešli celý graf v dostatočne krátkom čase. Ak sa podporný uzol nachádza v komunikačnom dosahu odosielateľa, informuje ho, že môže odoslať správy. Tie sú potom uložené niekde v podpornej štruktúre a keď sa prijímateľ dostane do vysielaceho dosahu podporného uzla, sú mu správy preposlané.

Výkonnosť tohto protokolu ovplyvňujú veľkosť siete, k a tvar podpornej štruktúry, ktorá má dve alternatívy:

- had - podporné uzly sa pohybujú tak, aby boli po dvojiciach prepojené. Vytvárajú reťaz k uzlov, ktorá sa pohybuje ako had. Hlava hada robí náhodný prechod grafom a ostatné uzly sledujú pohyb svojich predchodcov,
- každý podporný uzol robí náhodný prechod sieťou. Ak sa stretnutú dva podporné uzly, vymenia si všetky dáta, ktoré získali od posielateľov správ.

Modelovanie takýchto scenárií a informácie o pravdepodobnosti a čase stretnutia sa uzlov sú dobrým základom pre tvorbu efektívnejších komunikačných protokolov v mobilných ad hoc sieťach.

7.4 Kontrola zahltenia siete

V sensorovej sieti aj napriek snahe minimalizovať objem prenášaných dát je počet správ stále značný. Najmä pri broadcastovaní môže ľahko nastať zahltenie siete. Kľúčovým symptómom zahltenia v bezdrôtovej sensorovej sieti je degradácia kvality rádiového kanála, ktorá je spôsobená nárastom komunikačnej premávky v ostatných častiach siete. Pretože rádiové linky nie sú medzi sebou tienené ako v bežných sieťach, komunikácia traverzujúca akúkoľvek časť siete má škodlivý vplyv na kvalitu kanálov a stratovosť v ostatných častiach siete [74]. Pre zníženie týchto negatívnych vplyvov na

¹²motion-graph

sieť je v niektorých aplikáciách vhodné používať techniky na predchádzanie zahltenia.

Použitie ďalšieho protokolu, ktorý so sebou prináša zvýšené náklady na prevádzku siete, sa zdá byť neefektívnym krokom. Najmä v senzorových sieťach obmedzených nízkou výpočtovou silou a limitovanou energiou. Treba si ale uvedomiť, že pri nastatí zahltenia siete sa v danej časti väčšina správ zahadzuje a musia byť poslané nanovo. Keďže senzorové siete obsahujú veľké množstvo uzlov, ktoré sú nahusto rozmiestnené, je zahltenie bežným javom a teda počet zahodených správ a ich opätovné preposielanie výrazne vplýva na energetickú spotrebu siete. Pri použití jednoduchšej techniky, v ktorej si uzly budú signalizovať výskyt zahltenia tak môžeme ušetriť značné množstvo poslaných dát.

Hull a kol. [74] predstavili schému pre kontrolu zahltenia nazývanú *Fúzia*¹³. Skladá sa z troch techník aplikovaných na rôznych vrstvách štandardného vrstvomodelu siete, *hop po hope kontrola toku*¹⁴, *limitovanie rýchlosti* a *preferovanie u MAC vrstvy*.

Pri hop po hope kontrole toku každý uzol do hlavičky správ pridáva bit pre zahltenie, ktorý bude signalizovať nastatie zahltenia susedom. Vďaka broadcastovacej povahe senzorových sietí sa pomocou jedného bitu upozornia všetci susedia uzla a uzly získavajú informácie o zahltení pri prijatí každej správy. Hop po hope kontrola toku pozostáva z dvoch častí:

- *Detekcia zahltenia* - existujú dve techniky na detekovanie zahltenia. Prvá je založená na sledovaní veľkosti čakacieho radu senzora. Ak veľkosť voľného miesta v rade pre odchádzajúce správy v senzore presiahne určitú hodnotu α , uzol nastaví bit pre zahltenie v každej odchádzajúcej správe, v opačnom prípade ho vynuluje. Druhá technika je založená na vzorkovaní kanála, ktoré zaviedol algoritmus CODA [75]. Ak paket čaká na odoslanie, senzor začne kontrolovať stav komunikačného kanála v pevne daných intervaloch. V závislosti od počtu vzoriek, v ktorých je kanál zaneprázdnený, uzol vypočíta faktor jeho používania. Ak tento faktor presiahne určitú hranicu, v každej odchádzajúcej správe sa nastavuje bit pre zahltenie, inak je nulový.
- *Vyhýbanie sa zahlteniu* - je technika, pri ktorej uzly škrtia svoju komunikáciu aby zabránili pretečeniu čakacích radov (a tým zahodeniu správy) u svojich 1-hop susedov. Ak uzol príjme správu s nastaveným zahlcovacím bitom, prestane vysielat' pakety, aby umožnil zmenšenie radov u svojich susedov. Pri permanentnom zahltení sa týmto spô-

¹³fusion

¹⁴hop-by-hop flow control

sobom dostane spätná väzba až k zdroju dát, čím môže spôsobiť, že kontrola toku dát na aplikačnej úrovni zníži frekvenciu posielania údajov zo zdroja. Problémom je, že uzol po prijatí správy s bitom pre zahltenie prestane vysielat', a tak nemôže upozorniť svojich susedov na výskyt zahltenia. Preto je po signalizácii zahltenia povolená ešte jedna správa, ktorá naň upozorňuje.

Pri zväčšujúcej sa veľkosti siete je zahadzovanie paketov stále drahšie. Zahodené pakety musia byť preposlané ešte raz a pri veľkých vzdialenostiach spotrebúvajú energiu viacerým uzlom. Pri limitovaní rýchlosti, každý uzol počúva komunikáciu svojho otca, aby určil počet unikátnych zdrojov (N), ktoré cez neho routujú svoje pakety. Každý uzol disponuje počítadlom, ktoré obmedzuje množstvo odosielaných správ z uzla. Uzol môže odoslať správu len vtedy, keď jeho počítadlo je nenulové a pri každej odoslanej správe sa zníži o jednotku. Naopak ak uzol zistí, že jeho otec preposlal N správ, zvýši počítadlo (to má určené aj maximálnu hodnotu). Takýmto spôsobom je zaručené, že každý uzol vysielala približne takou rýchlosťou ako jeho potomci.

V niektorých prípadoch je potrebné odstraňovať zahltenie už v nižších vrstvách modelu. Na to slúži technika náhodného pozdržania odoslania správy u uzla [76]. Ak je senzor zahltený, jeho čas pozdržania bude len štvrtinový, oproti času pre nezahltený uzol. Zahltený uzol tak bude mať väčšie šance dostať sa k médiu a zmierniť svoje zaťaženie.

Posledný stupeň pre kontrolu zahltenia je na aplikačnej úrovni, ktorá by mala reagovať na permanentné zahltenia zmenou rýchlosti posielania dát.

Ďalším príkladom schémy na kontrolu zahltenia je *CODA* (Congestion detection and avoidance) [75]. Obdobne ako pri fúzii, aj tu sa stretávame s aplikovaním viacerých techník na dosiahnutie požadovaného výsledku:

- *Detekcia zahltenia* - ako už bolo vyššie spomenuté, *CODA* na zistenie zahltenia používa techniku vzorkovania komunikačného kanála. Po zistení zahltenia ho uzly signalizujú svojim susedom pomocou mechanizmu spätnej väzby.
- *Otvorená slučka, hop po hope spätná väzba* - pri zistení zahltenia uzly posielajú správy spätnej väzby smerom k zdroju signálu. Uzol po prijatí správy so spätnou väzbou buď priškrtní odosielanie správ alebo zahodí niektoré pakety. To závisí od jeho lokálnej politiky zahltenia. Uzly, ktoré sú v smere zdroja správ (nachádzajú sa na routovacej ceste od zdroja) sa po prijatí správy so spätnou väzbou na základe ich lokálneho stavu siete rozhodnú, či ju budú preposielat' ďalej.

- *Uzavretá slučka, regulácia viacerých zdrojov* - ak rýchlosť, ktorou posiela zdroj údaje je nižšia ako určitý zlomok maximálnej teoretickej priepustnosti kanála, začne sa regulovať. Pri prekročení tejto hranice sa pravdepodobne zdroj bude podieľať na vytvorení zahltenia, a preto sa sputí uzavretá slučka pre kontrolu zahltenia. V tomto stave zdroj požaduje konštantnú spätnú väzbu v krátkych časových intervaloch od sink-u. Vytvorí sa tak slučka medzi sink-om a zdrojom, aby zdroj vedel určiť rýchlosť posielania dát, ktorá nepovedie k zahlteniu.

Kvôli broadcastovacej povahe senzorových sietí a zdieľaniu komunikačného kanála sú bezdrôtové ad hoc senzorové siete náchylné na vznik zahltenia, a tým aj straty paketov. Preto predchádzanie takýmto situáciám pomôže zefektívniť výkon siete, ale aj ušetriť nemalé energetické zdroje. Ako vidno, problém zahltenia sa dotýka každej vrstvy referenčného modelu, a preto by sa mal brať do úvahy pri návrhu všetkých protokolov a nielen protokolov pre sieťovú vrstvu.

Kapitola 8

Lokalizácia

Lokalizácia je proces, pomocou ktorého sa určuje pozícia uzlov v sensorových ad hoc sieťach. Poznať pozíciu uzlov v sieti je potrebné pre zabezpečenie fungovania rôznych aspektov siete ako je geografické routovanie, zameriavanie a sledovanie objektov a detekcia udalostí s ich pozíciou. V niektorých prípadoch sa priestorové koordináty používajú aj na pomenovanie nasnímaných dát. Lokalizácia sa dá dosiahnuť manuálnou konfigúraciou siete, alebo prídavným hardvérom v uzloch ako je globálny polohovací systém (GPS). Prvá možnosť okrem malého počtu sietí nepripadá do úvahy kvôli ad hoc nasadzovaniu sensorových sietí. V neprospech druhej hovorí cena GPS zariadení a ich energetická náročnosť. Navyše GPS prijímanie vyžaduje priame spojenie medzi zariadením a satelitom, čo znižuje jeho účinnosť v budovách a pri výskyte prírodných prekážok ako je bujná vegetácia alebo hornatá krajina. V sensorových sieťach sa preto na určovanie koordinátov sensorových uzlov používajú rôzne polohovacie algoritmy. Väčšinou ide o schému, v ktorej si uzly zmerajú vzdialenosť k svojim susedom a z dostupných informácií vypočítajú svoju polohu.

8.1 Výpočtová zložitosť lokalizácie

Za lokalizačný problém považujeme problém rekonštrukcie pozícií množiny sensorov ak máme k dispozícii vzdialenosti pre každú dvojicu uzlov, pre ktorú platí, že každý z dvojice sa nachádza v jednotkovom kruhu s polomerom r toho druhého. Niektoré z uzlov môžu byť *majáky (kotvy)*. Sú to uzly, pre ktoré je pozícia známa. Pre zjednodušenie a vyhnutie sa iracionálnym číslam sa predpokladá, že problém je prezentovaný so štvorcami vzdialeností. Lokalizačný problém je podobný problému pre vnorenie grafu. Zistiť, či ohodnotený graf sa dá vnoriť do k -rozmerného priestoru, je však NP-ťažký

problém [77]. Ide ale o všeobecné grafy.

V sensorovej sieti môžu uzly komunikovať len s uzlami, ktoré sú v ich komunikačnom dosahu. Grafy formované linkami len medzi uzlami, ktoré si dokážu zmerať vzdialenosť medzi sebou, sa lepšie modelujú jednotkovými kruhovými grafmi. Toto obmedzenie dovoľuje dizajnovanie výpočtovo efektívnych lokalizačných algoritmov. Lokalizácia v sieti sa dá formulovať ako semidefinitný programovací problém, a teda sa môže efektívne riešiť [78]. Predpokladom ale je, že ide o husté grafy (s veľkým počtom hrán). Algoritmus vyžaduje, aby aspoň $\Omega(n^2)$ dvojíc uzlov poznalo svoju relatívnu vzdialenosť, kde n je počet uzlov v sieti. Vo všeobecnej sieti však za určitých podmienok stačí, keď $O(n)$ dvojíc uzlov bude poznať svoju vzdialenosť [79]. Pre riedke grafy je však situácia oveľa horšia. Lokalizácia v riedkych sieťach je NP-ťažká [80]. Toto zistenie je v kontraste s možnosťou lokalizácie v hustých grafoch a vysokou pravdepodobnosťou pre náhodné geometrické grafy.

Zo známych výsledkov teda vyplývajú niektoré možnosti pre lokalizáciu v sensorových sieťach:

- nájdenie aproximatívneho algoritmu pre riešenie lokalizácie v riedkych sieťach,
- keďže zložitosť problému závisí od hustoty uzlov v sieti, nájdenie hranice pre riešiteľnosť lokalizácie a nasadzovanie sensorových sietí s požadovanou hustotou,
- určenie takého komunikačného dosahu sensorov, aby sensorová sieť tvorila hustý graf alebo nasadenie malého počtu majákov s veľkým dosahom.

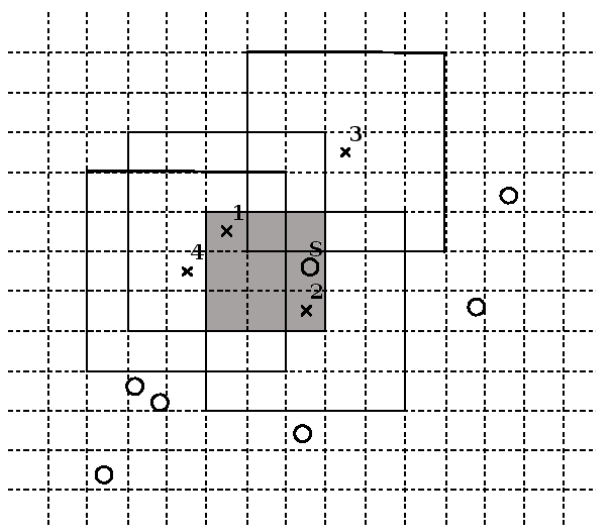
8.2 Lokalizácia pomocou majákov

Dizajn lokalizačného systému je výrazne ovplyvnený požiadavkami aplikácie ako sú presnosť polohovania alebo odhadovanie pozície v reálnom čase. Lokalizačné systémy založené na princípe majákov môžeme rozdeliť do dvoch skupín [81]. Na *tesne spárované systémy*¹ a na *voľne spárované systémy*².

Pevne spárované systémy obsahujú majáky, ktoré sú umiestnené na fixných pozíciách a sú káblom prepojené s centrálnym ovládačom. Vyznačujú sa síce vysokou presnosťou a splňajú požiadavky na sledovanie v reálnom

¹tightly coupled systems

²loosely coupled systems



Obrázok 8.1: Lokalizovanie prienikom

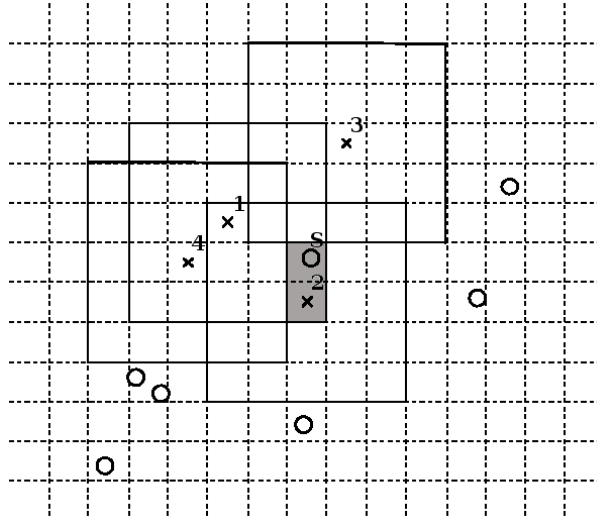
Štyri majáky (očíslované krížiky) a niekoľko ďalších uzlov (krúžky). Maják x broadcastuje svoju pozíciu (i, j) vo štvorci $R_{(i,j)}^\rho$, so stredom v (i, j) a dĺžkou strany $2\rho + 1$. Uzol S používa lokalizáciu prienikom. Svoju pozíciu určuje v prieniku majákov 1 a 2 (šedá oblasť).

čase, ale majú aj svoje nevýhody. Centralizácia odhadovania pozície zneumožňuje simultánne sledovanie viacerých zariadení. Ďalším obmedzením je fixná pozícia a káblové spojenie s majákmi, čo znemožňuje ich všestranné použitie.

Voľne spárované systémy obsahujú majáky, ktoré sú bezdrôtové a kompletne decentralizované. Ľahko sa nasadzujú a môžu byť použité vo veľkých množstvách. Táto výhoda je ale na úkor presnosti, ktorú dosahujú pevne spárované systémy. Ďalším problémom je, že pri ich náhodnom rozmiestnení nemusia dosiahnuť dobré pokrytie a kvôli chýbajúcej centrálnej kontrole medzi nimi neexistuje explicitná koordinácia a môže dochádzať k interferencii ich signálov.

Algoritmy pre lokalizáciu sa zameriavajú hlavne na dva problémy. *Odhadovanie pozície a optimálne umiestňovanie uzlov.*

Ako majáky sa väčšinou používajú rovnaké uzly ako sú v sieti, len sú vybavené GPS zariadením. Pomocou GPS uzol cez satelit zisťuje svoju presnú pozíciu a túto informáciu potom broadcastuje do siete. Ostatné uzly pomocou tejto informácie a vypočítanej vzdialenosti medzi sebou a susedmi určujú vlastné pozície.



Obrázok 8.2: Lokalizovanie dohodou

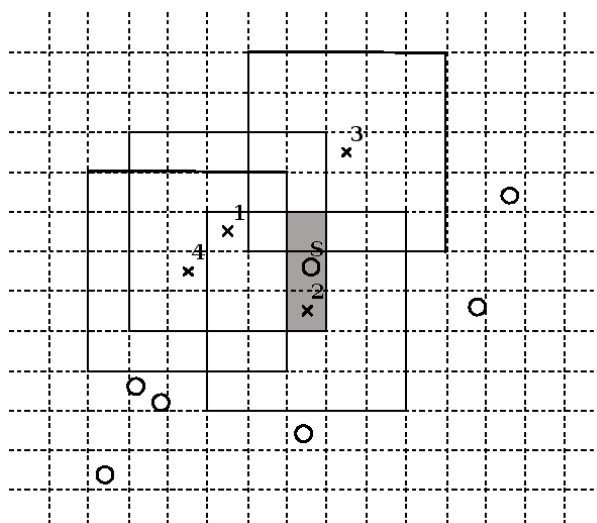
Uzol S používa lokalizáciu dohodou. Pozíciu určuje v prieniku susedných majákov 1 a 2, ale mimo dosahu ostatných majákov 3 a 4 (šedá oblasť).

Predpokladajme, že N senzorov je náhodne rozmiestnených v mriežke $[n + 1] \times [n + 1]$ a že len podmnožina týchto senzorov $K < N$ (majáky) pozná svoju pozíciu v zmysle súradníc bunky mriežky, v ktorej sa nachádzajú. Nech každý z K majákov broadcastuje svoju pozíciu, pričom broadcastovanie prijmu len uzly v komunikačnom dosahu majáka ρ , ktorý je definovaný ako štvorec $[2\rho + 1] \times [2\rho + 1]$ so stredom v majáku [82]. Za susedné uzly budeme považovať tie, ktoré sú schopné medzi sebou priamo komunikovať.

Najjednoduchší prístup pre senzor je zistiť pozície od všetkých svojich susedných majákov a určiť svoju pozíciu niekde v prieniku komunikačných dosahov týchto majákov (obrázok 8.1, podľa [82]). Ide o lokalizáciu *prienikom* [83]. Nech S je náhodne vybraný uzol a S_{k_1}, \dots, S_{k_m} sú jeho susedné majáky so súradnicami (x_i, y_i) . Nech A_S je množina buniek, ktoré sú možnými pozíciami pre S v danom protokole (tu prienik komunikačných dosahov susedných majákov). Ak S nemá susedné majáky, potom preňho ako možné pozície prichádzajú do úvahy všetky bunky mriežky:

$$A_S = \begin{cases} \bigcap_{i=1}^m R_{(x_i, y_i)}^\rho & \text{ak } m > 0 \\ Q_\rho & \text{inak} \end{cases}$$

Potom bunka $(i, j) \in R_S^{2\rho}$ patrí do A_S , ak žiaden z K majákov nepatrí do



Obrázok 8.3: Aproximácia lokalizovania dohodom

Uzol S používa aproximáciu lokalizácie dohodom. Pozíciu určuje v prieniku susedných majákov 1 a 2, ale mimo dosahu majáka 4 (šedá oblasť). Maják 3 sa ignoruje, lebo nie je susedom ani jedného zo susedov S .

oblasti $R_S^\rho \setminus R_{(i,j)}^\rho$.

Tento protokol môžeme vylepšiť, ak budeme predpokladať, že uzol pozná pozíciu všetkých K majákov a nielen tých, s ktorými susedí. Ide o lokalizáciu *dohodom*. Informácie o ostatných majákoch môže uzol získať použitím zaplavovania. Pozícia sa potom volí v prieniku komunikačných dosahov susedných majákov uzla, pričom nie je v dosahu žiadneho iného majáka (obrázok 8.2, podľa [82]). Opäť máme dané S a S_{k_1}, \dots, S_{k_m} , pričom $S_{k_{m+1}}, \dots, S_{k_K}$ sú ostatné majáky. Potom

$$A_S = \begin{cases} \bigcap_{i=1}^m R_{(x_i, y_i)}^\rho \setminus \bigcup_{i=m+1}^K R_{(x_i, y_i)}^\rho & \text{ak } m > 0 \\ Q_\rho \setminus \bigcup_{i=m+1}^K R_{(x_i, y_i)}^\rho & \text{inak} \end{cases}$$

Bunka $(i, j) \in R_S^{2\rho}$ patrí do A_S , ak žiaden z K majákov nepatrí do oblasti $(R_S^\rho \cup R_{(i,j)}^\rho) \setminus (R_S^\rho \cap R_{(i,j)}^\rho)$.

Pre dobrú aproximáciu lokalizácie dohodom predpokladajme, že majáky môžu počúvať broadcastovanie ostatných majákov. Najprv môžu broadcastovať len svoju polohu. Po tomto kroku môžu získať polohu svojich susedných

majákov a broadcastovať ju spolu so svojou polohou. Uzol si tak môže vytvoriť dva disjunktné zoznamy. Zoznam svojich susedov a zoznam susedov svojich susedov, ktorý nie sú zároveň aj jeho susedmi. Takto môže uzol zvoliť svoju pozíciu v prieniku dosahov svojich susedných majákov, pričom nie je v dosahu majákov, ktoré sú susedmi jeho susedných majákov (obrázok 8.3, podľa [82]). Opäť máme množinu S_{k_1}, \dots, S_{k_m} susedných majákov a množinu $S_{k_{m+1}}, \dots, S_{k_{m+l}}$ majákov, ktoré sú susedmi aspoň jedného z S_{k_1}, \dots, S_{k_m} majákov. Potom

$$A_S = \begin{cases} \bigcap_{i=1}^m R_{(x_i y_i)}^\rho \setminus \bigcup_{i=m+1}^{m+l} R_{(x_i y_i)}^\rho & \text{ak } m > 0 \\ Q_\rho & \text{inak} \end{cases}$$

Pre bunku $(i, j) \in R_S^{2\rho}$, jej príslušnosť do A_S závisí od počtu a pozície majákov v oblasti $R_{ij}^\rho \cap R_S^\rho$. Pre dostatočne veľké K je veľkosť oblasti možnej pozície uzla (oblasť neistoty) vrátená aproximáčnym protokolom s veľkou pravdepodobnosťou rovná veľkosti takejto oblasti vrátenej pôvodným protokolom pre lokalizáciu dohodou [82].

8.3 Nasadzovanie majákov

Poloha majákov a ich počet v sieti výrazne ovplyvňuje presnosť súradníc vypočítaných uzlami. Po nasadení majáka je potrebná jeho konfigurácia (zadanie jeho presných geografických koordinátov). V exteriéroch sa používa GPS zariadenie, ktorým je maják vybavený, v interiéroch sa väčšinou vytvára štruktúra pre nasadzovanie, a tak sú súradnice pre majáky dopredu známe. Pár majákom sa nastavujú súradnice manuálne a ostatné si ich odvodujú zo štruktúry v akej sú usporiadané (napríklad ak tvoria mriežku). Najlepším usporiadaním je zrejme uniformné nasadenie s veľkou hustotou. Práve táto možnosť však pre senzorové siete nepripadá do úvahy. Techniky pre nasadzovanie majákov sa pre riedke a husté siete navzájom líšia.

V sieťach s malou a strednou hustotou ad hoc voľba pozícií pre majáky nie je postačujúca na dosiahnutie dobrej lokalizácie v celej sieti. Algoritmus HEAP [81] používa inkrementálny prístup pri nasadzovaní. Je založený na myšlienke, že je lepšie objaviť vhodné miesta pre nasadenie malého počtu majákov tak, aby sa maximalizovalo zlepšenie lokalizácie ako znova nasadiť všetky majáky. HEAP predpokladá senzorovú sieť zostavenú z nasledujúcich entít: uzly, majáky a nasadzovače. Funguje nasledovne:

- Majáky si vymieňajú informácie medzi sebou a snažia sa zvoliť kandidátov vhodných pre umiestnenie nových majákov vo svojom susedstve

(v nejakom regióne s polomerom r). Vyberú si jednu pozíciu a pošlú ju svojmu otcovi (predpokladáme vytvorenú routovaciu štruktúru k nasadzovaču).

- Uzly prijímajú kandidátov od všetkých svojich susedných majákov a svojich detí. Môže sa využiť aj agregácia dát. Zvolia si jedného z kandidátov a pošlú ho svojmu otcovi.
- Nasadzovač získa kandidátov od svojich susedných majákov a detí. Odstráni všetkých kandidátov, ktorí nespĺňajú obmedzenia a zvolí vhodné polohy pre nasadenie nových majákov.

Pre siete s vysokou hustotou predstavujú Bulusu a kol. algoritmus STROBE³ [81]. STROBE začína s veľmi hustým nasadením majákov v sieti. Namiesto vyberania vhodných miest pre majáky v sieti ich nasadí vo veľkom počte, a potom z nich vyberá aktívnu množinu (zapínaním a vypínaním majákov), aby predĺžil ich životnosť. Jeho cieľom je, aby majáky kooperatívne dosahovali takú hustotu, pri ktorej sa nezníži presnosť lokalizácie. Teda snaží sa o dosiahnutie uniformnej lokalizačnej presnosti v celej sieti a počas celej jej životnosti a o maximalizovanie životnosti siete minimalizovaním spotreby energie.

Každý maják môže byť v jednom z troch stavov: *voliaci*, *označený* a *spiaci*. Maják začína v stave voliaci, v ktorom zapne rádio a broadcastuje svoju pozíciu každých T_B sekúnd a prijíma správy s pozíciami od svojich susedných majákov. Keď maják vchádza do stavu voliaci, nastaví si časovač na T_V sekúnd. Po jeho uplynutí sa na základe dostupných informácií rozhodne či prejde do stavu spiaci. Ak áno, informuje o tom broadcastovaním a prechádza do stavu spiaci. Inak prechádza do stavu označený. Maják sa zo spiaceho stavu prebúdz po uplynutí T_S sekúnd a prechádza do stavu voliaci. V stave označený, maják po dobu T_D v intervaloch T_B broadcastuje svoju pozíciu, a potom prechádza do stavu voliaci.

Rozlíšenie stavov voliaci a označený je kvôli zníženiu nákladov na prevádzku algoritmu, lebo v stave označený maják len odosiela správy ale v stave voliaci ich aj prijíma, aby sa mohol rozhodovať. Celkovú spotrebu energie a životnosť systému ovplyvňujú parametre T_V , T_D a T_S .

Lokalizácia je v sensorových sieťach nevyhnutnosťou pre fungovanie sieťových protokolov a pre plnenie cieľov niektorých aplikácií. Keďže vybavenie každého sensorového uzla polohovacím zariadením je drahé ako v zmysle ceny hardvéru, tak aj v zmysle energetickej náročnosti, používajú sa polohovacie

³selectively turning off beacons

algoritmy a len malá množina špeciálnych uzlov, ktoré poznajú svoju pozíciu. Ako bolo vyššie spomenuté, neexistuje lokalizačný algoritmus použiteľný všeobecne, dokonca pri niektorých sieťach je problém lokalizácie NP-ťažký. Obdobne problém nasadzovania majákov v sieti zatiaľ nemá uspokojivé riešenie. Lokalizácia tak ostáva ďalším z rady problémov, s ktorými sa potýkajú senzorové siete, a ktoré budeme musieť vyriešiť aby sme mohli naplno využiť ich potenciál.

Kapitola 9

Záver

Myšlienka senzorových sietí je pomerne stará. Zatiaľ čo pri svojom vzniku bola len v teoretickej rovine, rozvojom dnešných technológií sa dostávame ku konkrétnym sieťam a aplikáciám a sme nútení riešiť praktické problémy pri implementácii senzorových sietí. Za posledné desaťročie sa intenzívne skúmali rôzne aspekty ad hoc bezdrôtových sietí. Keďže existuje málo prototypov senzorových uzlov a väčšina z existujúcich je úzko špecializovaná a navrhnutá pre určitý typ siete, alebo dokonca pre konkrétnu aplikáciu, nevenovali sme sa v tejto práci problematike spodných vrstiev referenčného modelu, ktoré sú úzko späté s hardvérom.

Namiesto toho sme ponúkli zhrnutie najdôležitejších poznatkov a techník pre dizajn protokolov stredných vrstiev ako je linková a sieťová vrstva. V práci tak možno nájsť ucelený prehľad spôsobov a metodík na riešenie tradičných sieťových problémov. Keďže ide o rozsiahlu oblasť s množstvom problémov a veľkým množstvom literatúry, do práce sme zahrnuli len reprezentatívnu vzorku problémov a výsledkov získaných pri ich riešení. Pri jednotlivých problematikách sme poukázali aj na motiváciu, ktorá viedla k zvoleniu daného prístupu k ich riešeniu. Našou snahou bolo upozorniť aj na teoretickú časť problému, a tým získať jeho abstrakciu. Napriek tomu, že existuje veľa návrhov na riešenie problémov týkajúcich sa senzorových sietí, vo väčšine prípadov ide o riešenia vhodné len pre konkrétny typ aplikácie, postrádajúce všeobecnosť.

Výskum v oblasti senzorových sietí sa nachádza na polceste medzi prvotnou myšlienkou a ich hromadným nasadením v našom každodennom živote. Máme jasne pomenované problémy a výzvy, ktoré musíme zvládnuť. Doterajšie výsledky v tejto oblasti dávajú uspokojujúce závery, ale za cenu predpokladov a zjednodušení, ktoré sú v reálnom svete prinajmenšom ťažko dosiahnuteľné. Z týchto dôvodov vidíme bezdrôtové senzorové siete ako mladú neustále sa rozvíjajúcu oblasť, ktorá si zaslúži našu plnú pozornosť.

Literatúra

- [1] C. Y. Chong, S. P. Kumar: Sensor Networks: Evolution, opportunities, and challenges, Proceedings of the IEEE 91 (8), Aug. 2003
- [2] C. E. Nishimura, D. M. Conlon: IUSS dual use: Monitoring whales and earthquakes using SOSUS, Mar. Technol. Soc. J., vol. 27, no. 4, pp. 13–21, 1994
- [3] R. Rashid, G. Robertson: Accent: A communication-oriented network operating system kernel, In: Proc. 8th Symp. Operating System Principles, 1981, pp. 64–75
- [4] R. Rashid, D. Julin, D. Orr, R. Sanzi, R. Baron, A. Forin, D. Golub, M. Jones: Mach: A system software kernel, In: 34th Computer Society Int. Conf. (COMPCON), San Francisco, CA, 1989
- [5] C. Myers, A. Oppenheim, R. Davis, W. Dove: Knowledge-based speech analysis and enhancement, In: Int. Conf. Acoustics, Speech and Signal Processing, San Diego, CA, 1984
- [6] C. Y. Chong, S. Mori, K. C. Chang: Distributed multitarget multisensor tracking, In: Multitarget Multisensor Tracking: Advanced Applications, Y. Bar-Shalom, Ed. Norwood, MA: Artech House, 1990, pp. 247–295
- [7] R. T. Lacoss: Distributed mixed sensor aircraft tracking, In: Amer. Control Conf., Minneapolis, MN, 1987
- [8] D. S. Alberts, J. J. Garska, F. P. Stein: Network centric warfare: Developing and leveraging Information Superiority, Library of Congress Cataloging-in-Publication Data, ISBN 1-57906-019-6
- [9] The cooperative engagement capability: [Online] Available: <http://techdigest.jhuapl.edu/td1604/APLteam.pdf>

- [10] S. Kumar, D. Shepherd: SensIT: Sensor information technology for the warfighter, In: Proc. 4th Int. Conf. on Information Fusion, 2001, pp. TuC1-3–TuC1-9
- [11] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci: A Survey on Sensor Networks, *Communications Magazine*, IEEE, 40(8), Aug 2002, pp. 102–114
- [12] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, A. Chandrakasan: Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks, In: Proc. ACM MobiCom '01, Rome, Italy, July 2001, pp. 272–286
- [13] J. M. Kahn, R. H. Katz, K. S. J. Pister: Mobile networking for smart dust, In: Proc. ACM/IEEE Int. Conf. Mobile Computing and Networking (MobiCom), 1999, pp. 271–278
- [14] S. E. A. Hollar: COTS Dust, Msc. thesis, Berkley
- [15] J. Hill, D. Culler: A wireless embedded sensor architecture for system-level optimization, Technical report, U. C. Berkeley, 2001
- [16] K. Prasanna: Algorithm design and optimization for sensor systems, In: Proc. 1st Int. Work. Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004), Turku, Finland, 2004, pp. 1–2
- [17] V. Claesso, H. Lönn, N. Suri: Efficient TDMA synchronization for distributed embedded Systems, 20th Symposium on reliable distributed systems (SRDS), pp. 198–201, October 2001
- [18] T. Ojanperä, R. Prasad: Wideband CDMA for third generation mobile communications, Artech House, Incorporated, 1998
- [19] Wireless LAN medium access control (MAC) and physical layer (PHY) specification, IEEE Std. 802.11-1999 edition
- [20] E. Callaway, V. Bahl, P. Gorday, J. A. Gutierrez, L. Hester, M. Naeve, B. Heile: Home networking with IEEE 802.15.4: A developing standard for low-rate wireless personal area networks, In: IEEE Communications MAGAZINE, Special issue on Home Networking, August 2002, pp. 70–77
- [21] P. G. Spirakis: Algorithmic and foundational aspects of sensor systems, In: Proc. 1st Int. Work. Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004), Turku, Finland, 2004, pp. 3–8

- [22] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, K. Pister: System architecture directions for networked sensors, In: Proc. 9th Int. Conf. Architectural support for programming languages and operating systems, Cambridge, Massachusetts, United States, 2000, pp. 93–104
- [23] D. D. Hwang, B.-Ch. Ch. Lai, I. Verbauwhede: Energy-memory-security tradeoffs in distributed sensor networks, In: Proc. Ad-Hoc, mobile and wireless networks (ADHOC-NOW 2004), LNCS 3158, Springer, 2004
- [24] Ch. Karlof, N. Sastry, D. Wagner: TinySec: a link layer security architecture for wireless sensor networks, In: Proc. 2nd Int. Conf. Embedded networked sensor systems (SenSys 2004), November 2004
- [25] H. Dai, M. Neufeld, R. Han: ELF: an efficient log-structured flash file system for micro sensor nodes, In: Proc. 2nd Int. Conf. Embedded networked sensor systems (SenSys 2004), November 2004
- [26] S. Madden, M. J. Franklin, J. Hellerstein, W. Hong: TAG: A tiny aggregation service for ad-hoc sensor networks, In: Proc. 5th Annual Symposium on Operating Systems Design and Implementation (OSDI), 2002
- [27] Y. Yao, J. Gehrke: The Cougar approach to in-network query processing, ACM SIGMOD Record, vol. 31, pp. 9, 2002
- [28] J. Zhao, R. Govindan: Understanding packet delivery performance in dense wireless sensor networks, In: Proc. 1st Int. Conf. Embedded networked sensor systems (SenSys 2003), November 2003
- [29] M. Achir, L. Ouvry: Probabilistic model for energy estimation in wireless sensor networks, In: Proc. 1st Int. Work. Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004), Turku, Finland, 2004, pp. 157–170
- [30] W. Ye, J. Heidemann, D. Estrin: Medium access control with coordinated adaptive sleeping for wireless sensor networks, IEEE/ACM Transactions on Networking (TON), 12 (3)
- [31] A. El-Hoiydi, J. D. Decotignie: WiseMAC: An ultra low power MAC protocol for multi-hop wireless sensor networks, In: Proc. 1st Int. Work. Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004), Turku, Finland, 2004, pp. 18–31
- [32] J. Polastre, J. Hill, D. Culler: Versatile low power media access for wireless sensor networks, In: Proc. 2nd Int. Conf. Embedded networked sensor systems (SenSys 2004), November 2004

- [33] H. Zhang, J. C. Hou: Maintaining sensing coverage and connectivity in large sensor networks, technical report, UIUCDCS-R-200302351, UIUC, June 2003
- [34] J. Jiang, W. Dou: A Coverage-Preserving density control algorithm for wireless sensor networks, In: Proc. Ad-Hoc, mobile and wireless networks (ADHOC-NOW 2004), LNCS 3158, Springer, 2004
- [35] S. Funke, A. Kesselman, Z. Lotker, M. Segal: Improved approximation algorithms for connected sensor cover, In: Proc. Ad-Hoc, mobile and wireless networks (ADHOC-NOW 2004), LNCS 3158, Springer, 2004
- [36] P. Bose, A. Maheshwari, P. Morin, J. Morrison: The grid placement problem, In: Proc. 7th Annual Work. Algorithms and data structures, vol. 2125 of LNCS, pp. 180–191, 2001
- [37] R. Merz, J.-Y. Le Boudec, J. Widmer, B. Radunović: A Rate-adaptive MAC protocol for low-power ultra-wide band ad-hoc networks, In: Proc. Ad-Hoc, mobile and wireless networks (ADHOC-NOW 2004), LNCS 3158, Springer, 2004
- [38] S.-H. Lee, J.-M. Choi, Y.-B. Ko: UAMAC: Unidirectional-link aware MAC protocol for heterogeneous ad hoc networks, In: Proc. Ad-Hoc, mobile and wireless networks (ADHOC-NOW 2004), LNCS 3158, Springer, 2004
- [39] S. Ganeriwal, R. Kumar, M. B. Srivastava: Timing-sync protocol for sensor networks, In: Proc. 1st Int. Conf. Embedded networked sensor systems (SenSys 2003), November 2003
- [40] R. Cristescu, B. Beferull-Lozano, M. Vetterli: On network correlated data gathering, In: 23rd Conf. IEEE Communications Society (INFOCOM), 2004
- [41] S. Zou, I. Nikolaidis, J. J. Harms: Efficient data collection trees in sensor networks with redundancy removal, In: Proc. Ad-Hoc, mobile and wireless networks (ADHOC-NOW 2004), LNCS 3158, Springer, 2004
- [42] M. Enachescu, A. Goel, R. Govindan, R. Motwani: Scale free aggregation in sensor networks, In: Proc. 1st Int. Work. Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004), Turku, Finland, 2004, pp. 71–84

- [43] S. Nath, P. B. Gibbons, S. Seshan, Z. R. Anderson: Synopsis diffusion for robust aggregation in sensor networks, In: Proc. 2nd Int. Conf. Embedded networked sensor systems (SenSys 2004), November 2004
- [44] N. Shrivastava, C. Buragohain, D. Agrawal, S.Suri: Medians and beyond: New aggregation techniques for sensor networks, In: Proc. 2nd Int. Conf. Embedded networked sensor systems (SenSys 2004), November 2004
- [45] B. Hong, V. K. Prasanna: Constrained flow optimization with applications to data gathering in sensor networks, In: Proc. 1st Int. Work. Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004), Turku, Finland, 2004, pp. 187–200
- [46] D. Ganesan, B. Greenstein, D. Perelyubskiy, D. Estrin, J. Heidemann: An evaluation of multi-resolution storage for sensor networks, In: Proc. 1st Int. Conf. Embedded networked sensor systems (SenSys 2003), November 2003
- [47] B. Przydatek, D. Song, A. Perrig: SIA: secure information aggregation in sensor networks, In: Proc. 1st Int. Conf. Embedded networked sensor systems (SenSys 2003), November 2003
- [48] E. Elnahrawy, B. Nath: Poster abstract: online data cleaning in wireless sensor networks, In: Proc. 1st Int. Conf. Embedded networked sensor systems (SenSys 2003), November 2003
- [49] J. Kuruvila, A. Nayak, I. Stojmenovic: Progress based localized power and cost Aware routing algorithms for ad hoc and sensor wireless networks, In: Proc. Ad-Hoc, mobile and wireless networks (ADHOC-NOW 2004), LNCS 3158, Springer, 2004
- [50] F. Kuhn, R. Wattenhofer, Y. Zhang, A. Zollinger: Geometric ad-hoc routing: of theory and practice, In: Proc. 22nd annual symposium on Principles of distributed computing (PODC), 2003, pp. 63–72
- [51] C. H. Papadimitriou, D. Ratajczak: On a conjecture related to geometric routing, In: Proc. 1st Int. Work. Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004), Turku, Finland, 2004, pp. 9–17
- [52] A. Rao, C. Papadimitriou, S. Shenker, I. Stoica: Geographic routing without location information, In: Proc. 9th Annual Int. Conf. Mobile computing and networking, ACM Press (2003), pp. 96–108

- [53] K. K. Vadde, V. R. Syrotiuk: On timers of routing protocols in Manets, In: Proc. Ad-Hoc, mobile and wireless networks (ADHOC-NOW 2004), LNCS 3158, Springer, 2004
- [54] E. Chávez, S. Dobrev, E. Kranakis, J. Opatrny, L. Stacho, J. Urrutia: Route discovery with constant memory in oriented planar geometric networks, In: Proc. 1st Int. Work. Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004), Turku, Finland, 2004, pp. 147–156
- [55] E. Kranakis, H. Singh, J. Urrutia: Compass routing on geometric networks, In: Proc. 11th Canadian Conference on Computational Geometry, August 1999, pp. 51–54
- [56] X.-Y. Li: Algorithmic, geometric and graphs issues in wireless networks, Journal of Wireless Communications and Mobile Computing (WCMC), 2003, Wiley Publications, Volume 3, Issue 2, pp. 119–140
- [57] G. T. Toussaint: The relative neighborhood graph of a finite planar set, Pattern Recognition, vol. 12, no. 4, 1980, pp. 261–268
- [58] K. R. Gabriel, R. R. Sokal: A new statistical approach to geographic variation analysis, Systematic Zoology, vol. 18, 1969, pp. 259–278
- [59] X.-Y. Li, P.-J. Wan, Y. Wang: Power efficient and sparse spanner for wireless ad hoc networks, In: IEEE Int. Conf. Computer Communications and Networks (ICCCN01), 2001, pp. 564–567
- [60] P. Bose, P. Morin, I. Stojmenovic, J. Urrutia: Routing with guaranteed delivery in ad hoc wireless networks, ACM/Kluwer Wireless Networks, vol. 7, no. 6, 2001
- [61] M. Seddigh, J. S. Gonzalez, I. Stojmenovic: Rng and internal node based broadcasting algorithms for wireless one-to-one networks, ACM Mobile Computing and Communications Review, vol. 5, no. 2, 2002, pp. 37–44
- [62] X.-Y. Li, Y. Wang: Efficient construction of low-weighted bounded degree planar spanner, International Journal of Computational Geometry and Applications, World Science Publications, vol. 14, no. 1-2, April 2004, pp. 69–84
- [63] J. Gudmundsson, C. Levcopoulos, G. Narasimhan: Fast greedy algorithms for constructing sparse geometric spanners, SIAM Journal of Computing 31, 5 (2002), pp. 1479-1500

- [64] W. R. Heinzelman, J. Kulik, H. Balakrishnan: Adaptive protocols for information dissemination in wireless sensor networks, In: Proc. ACM MobiCom '99, Seattle, WA, 1999, pp. 174–185
- [65] S. Hedetniemi, A. Liestman: A survey of gossiping and broadcasting in communication networks, *Networks*, vol. 18, 1988
- [66] P. Sinha, R. Sivakumar, B. Vaduvur: Enhancing ad hoc routing with dynamic virtual infrastructures, In: Proc. IEEE INFOCOM (2001), pp. 1763–1772
- [67] G. Călinescu, I. I. Măndoiu, P.-J. Wan, A. Z. Zelikovsky: Selecting forwarding neighbors in wireless ad hoc networks, *Mobile Networks and Applications* 9 (2), April 2004, pp. 101–111
- [68] V. Chvátal: A greedy heuristic for the set-covering problem, *Mathematics of Operation Research* 4 (3), 1979, pp. 233–235
- [69] B. N. Clark, C. L. Colbourn, D. S. Johnson: Unit disk graphs, *Discrete Mathematics* 86 (1990), pp. 165–177
- [70] H. Bronnimann, M. T. Goodrich: Almost optimal set covers in finite VC-dimension, In: Proc. 10th ACM Symp. on Computational Geometry (SCG) (1994), pp. 293–302
- [71] Ch. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, F. Silva: Directed diffusion for wireless sensor networking, *IEEE/ACM Transactions on Networking (TON)*, 11 (1)
- [72] T. Dimitriou, S. Nikolettseas, P. Spirakis: Analysis of the information propagation time among mobile hosts, In: Proc. Ad-Hoc, mobile and wireless networks (ADHOC-NOW 2004), LNCS 3158, Springer, 2004
- [73] I. Chatzigiannakis, S. Nikolettseas, P. Spirakis: Distributed communication and control algorithms for ad-hoc mobile networks, In: *Journal of Parallel and Distributed Computing (JPDC)*, Special issue on mobile ad-hoc networking and computing, 63 (2003), pp. 58–74
- [74] B. Hull, K. Jamieson, H. Balakrishnan: Mitigating congestion in wireless sensor networks, In: Proc. 2nd Int. Conf. Embedded networked sensor systems (SenSys 2004), November 2004
- [75] Ch.-Y. Wan, S. B. Eisenman, A. T. Campbell: CODA: congestion detection and avoidance in sensor networks, In: Proc. 1st Int. Conf. Embedded networked sensor systems (SenSys 2003), November 2003

- [76] I. Aad, C. Castelluccia: Differentiation mechanisms for IEEE 802.11, In: Proc. IEEE INFOCOM Conf., Anchorage, AK, April 2001, pp. 209–218
- [77] J. B. Saxe: Embeddability of weighted graphs in k-space is strongly NP-hard, In: Proc. 17th Allerton Conf. in Communications, Control and Computing, 1979, pp. 480–489
- [78] Y. Shang, W. Ruml: Improved MDS-based localization, In: Proc. IEEE INFOCOM '04, Hong Kong, China, April 2004
- [79] T. Eren, D. Goldenberg, W. Whitley, Y. R. Yang, S. Morse, B. D. O. Anderson, P. N. Belhumeur: Rigidity, computation and randomization of network localization, In: Proc. IEEE INFOCOM '04, Hong Kong, China, April 2004
- [80] J. Aspnes, D. Goldenberg, Y. R. Yang: On the computational complexity of sensor network localization, In: Proc. 1st Int. Work. Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004), Turku, Finland, 2004, pp. 32–44
- [81] N. Bulusu, J. Heidemann, D. Estrin, T. Tran: Self-configuring localization systems: Design and experimental evaluation, ACM Transactions on Embedded Computing Systems (TECS), 3 (1)
- [82] G. Stupp, M. Sidi: The expected uncertainty of range free localization protocols in sensor networks, In: Proc. 1st Int. Work. Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004), Turku, Finland, 2004, pp. 85–97
- [83] S. Simic, S. Sastry: Distributed localization in wireless ad hoc networks, Technical Report UCB/ERL M02/26, UC Berkeley, 2002