Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Časová synchronizace v bezdrátových senzorických sítích

Plzeň 2019 Šteflová Tereza

Místo této strany bude zadání práce.

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracovala samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 25. dubna 2019

Šteflová Tereza

Poděkování

Ráda bych poděkovala vedoucímu práce, panu Ing. Jiřímu Ledvinovi, CSc., za odborné konzultace a vstřícné jednání ve všech situacích.

Abstract

This thesis is dedicated to the topic of time synchronization in wireless sensor networks. The main goal is to create an overview of algorithms that are being used for this purpose. The work is divided into three parts. The first part of the work describes the architecture and usage of wireless sensor networks and also their specifics and constraints. Next part deals with the topic of time synchronization in general, explains keywords and common algorithms for time synchronization. The main interest is to describe protocols of time synchronization in wireless sensor networks with accuracy, safety and energy consumption in mind.

Abstrakt

Tato bakalářská práce je zaměřena na metody časové synchronizace v bezdrátových senzorických sítích. Jejím cílem je zejména shrnutí algoritmů, které se k tomuto účelu používají. Práce je rozdělena na tři hlavní kapitoly. První je věnována popisu a využití bezdrátových senzorických sítí s důrazem na jejich specifické vlastnosti a omezení. V další části je nastíněn problém časové synchronizace obecně, jsou vysvětleny pojmy týkající se časové synchronizace a běžně užívané algoritmy k jejímu dosažení. Hlavní důraz je následně kladen na popis protokolů, které se používají k časové synchronizaci v bezdrátových senzorických sítích, a jejich zhodnocení z hlediska přesnosti, bezpečnosti a energetické náročnosti.

Obsah

1 Úvod

Bezdrátová senzorická síť je bezdrátové propojení autonomních zařízení, která spolu komunikují a označují se jako uzly. Uzly používají senzory pro monitorování různých fyzikálních veličin z okolí. Aby síť plnila svou funkci, je třeba zajistit koordinovanou činnost jejích součástí. Důležitým tématem z hlediska komunikace mezi uzly je časová synchronizace. Cílem této práce je analýza, shrnutí a zhodnocení metod časové synchronizace v bezdrátových senzorických sítích.

Úvodní kapitola se věnuje seznámení s bezdrátovými senzorickými sítěmi, popisuje jejich komponenty a architekturu. Uzly typicky obsahují komunikační a výpočetní část, senzory a zdroj energie. Dále jsou popsány typické vlastnosti senzorických sítí, kterým je nutno se při implementaci synchronizačních algoritmů přizpůsobit. Jsou to například omezený zdroj energie, nízká cena, škálovatelnost, bezpečnost a další. Důležitým požadavkem je časová synchronizace. Je klíčová pro správné a koordinované fungování senzorů, a tedy i celé sítě.

Třetí kapitola je obecným úvodem do časové synchronizace. Jsou zde vysvětleny druhy času, podle kterého lze síť synchronizovat, a také chyby, které je třeba synchronizací odstranit. Jsou zde definovány základní pojmy. Tato část poskytuje základní dělení metod časové synchronizace a popisuje některé známé synchronizační metody, používané např. i v počítačových sítích.

Další kapitola se věnuje popisu běžně užívaných synchronizačních protokolů pro bezdrátové sítě. U těchto protokolů je popsán postup synchronizace, další popis záleží na konkrétním typu protokolu. Týká se například struktury zasílaných zpráv, možných optimalizací daného protokolu atd.

V posledních dvou kapitolách probíhá hlavní zhodnocení a shrnutí vlastností vybraných synchronizačních algoritmů. Hodnotícím kritériem je přesnost a energetická náročnost, které se vzájemně ovlivňují a mají zpravidla protichůdné požadavky. U každého protokolu je uvedena přesnost synchronizace, které je možné za jeho použití dosáhnout. Dalším předmětem hodnocení je bezpečnost těchto protokolů. Je uvažována bezpečnost z hlediska vnitřních chyb sítě, jako je výpadek uzlu, i z pohledu vnějšího napadení. Jsou nastíněny metody obrany proti útokům na síť.

Závěrem jsou shrnuty výsledky celé práce a zejména předchozí kapitoly.

2 Základní charakteristika bezdrátových senzorických sítí

Bezdrátová senzorická síť (WSN, Wireless Sensor Network) je typ počítačové sítě, která je založena na velkém počtu spolupracujících uzlů. Interakce mezi nimi umožňují ovládat síť jako funkční celek. Příkladem využití WSN jsou různé aplikace pro automatizaci, monitorování (např. pohybu osob), pro sběr dat, která slouží k rozhodování, nebo i ke statistickým účelům. Nachází využití prakticky ve všech oblastech lidské činnosti.

Příkladem mohou být dále systémy pro detekci znečištění ovzduší, detekci oxidu uhelnatého, měření emisí továren. Také monitorovací sítě pro upozornění o lesním požáru, nebo při úniku radiace v jaderných elektrárnách; dále měření stability konstrukcí, kde jsou monitorovány vibrace a stav materiálu např. v mostech a budovách. Pro detekci smartphonů, které pracují s WiFi nebo Bluetooth rozhraním. S využitím WSN je možné měření hustoty dopravy nebo úrovně odpadu v kontejnerech za účelem optimalizace. Existuje celá řada dalších chytrých aplikací např. při hledání volného parkování, při nakupování a použitelných v domácnosti.

Senzorické sítě jsou navrhovány pro různá prostředí a konstruovány podle svého konkrétního účelu. Jsou využívány i v extrémních podmínkách pod zemí nebo pod vodou. Na tom je závislá konstrukce senzoru, software a způsob komunikace, hardwarové vybavení a výpočetní výkon. Pokud je například nízká četnost snímání a data jsou přenášena pouze při významných změnách, hodí se senzory s nižším výkonem a dlouhou výdrží. Pro návrh senzorických sítí neexistuje žádný obecný model, vždy je třeba vycházet z potřeb konkrétní aplikace, pro kterou je síť tvořena.

Pozn.: V textu je v různém kontextu zaměňován pojem uzel a senzor podle toho, zda je kladen důraz na jeho snímací funkci, nebo je brán jen jako část systému.

2.1 Komponenty sítě

WSN je tvořena uzly, jejich počet se pohybuje od několika jednotek až po tisíce. Uzly zastávají různé funkce, především sbírají data z okolí, zpracovávají je a komunikují mezi sebou. Každý uzel mapuje jemu přidělený úsek. Tyto je možné označit jako zdrojové uzly. Síť má zpravidla jeden nebo více uzlů, které fungují jako základnová stanice (base station, sink node). Ta například odděluje WSN od sítě uživatele, Internetu, nebo je to například počítač, který sbírá informace z více uzlů. Uzly mohou zastávat více funkcí, můžou snímat fyzikální podnět a zároveň ho zpracovávat.

Za další komponentu WSN by se dalo považovat komunikační médium. Tato část umožňuje výměnu informací mezi uzly. Bezdrátové senzorické sítě používají např. elektromagnetické, optické nebo ultrazvukové vysílače a přijímače.

2.1.1 Komponenty uzlu

Hlavními částmi uzlu senzorické sítě jsou mikrokontroler, bezdrátové komunikační zařízení, senzory, paměť a zdroj energie. Podle své funkce je dál vybaven např. klávesnicí, displejem apod. Důležitou součástí jsou i obvody pro šifrování komunikace.

Výpočetní jednotkou uzlu je mikrokontroler s procesorem, programovou a datovou pamětí. Mikrokontrolery používají Flash paměť pro uložení programu, RAM pro data používaná programem, EEPROM pro ukládání statických dat. Procesor je navržen tak, aby šetřil energií. Důsledkem toho a zároveň kvůli malým rozměrům uzlu je i nižší výpočetní výkon. Stálá velkokapacitní pamět slouží pro uložení dat, např. pokud uzly zpracovávají větší objemy měření nebo je třeba, aby data byla uložena i při výpadku napájení.

Senzory uzlů měří různé veličiny jako pohyb, tlak, teplotu, vlhkost nebo čas, podle své funkce. Uzly jsou navíc vybaveny převodníkem analogového signálu na číslicový (ADC), který je potřeba k tomu, aby analogový signál ze senzoru byl zpracovatelný jeho výpočetní jednotkou v digitální podobě.

Ke snímání fyzikálních jevů z okolí mohou sloužit také inteligentní MEMS (Micro-Electro-Mechanical Systems) čidla, která umožňují snímat např. zrychlení a Coriolisovu sílu, tlak, intenzitu magnetického pole, velikost a směr gravitačního pole. Také umožňují snímat chemické veličiny, např. přítomnost určitých látek ve vzduchu, kyselost apod.

Jednou z nejdůležitějších částí uzlu je zdroj energie. Je důležité, aby všechny ostatní komponenty uzlu měly co nejmenší spotřebu. Obvykle se využívají baterie, jsou ale možné i jiné způsoby napájení, např. fotovoltaické

panely, změny tlaku, termoelektrické články nebo vibrační zdroje energie, které převádí mechanické chvění na elektrickou energii. Uzly jsou nicméně ve většině případů napájeny zdrojem s omezenou kapacitou, baterií nebo akumulátorem. Pokrok v oblasti získávání energie z okolí znamená velkou výhodu, zejména na těžce přístupných místech, kde je složité doplňovat zdroj energie. Vzhledem k tomu, že uzly sítě mohou být běžně nepřístupné, je nutná i jejich autokonfigurace.

Nejen z hlediska časové synchronizace je důležité, aby každý uzel měl k dispozici čítač, který funguje jako hodiny uzlu. Může být součástí mikroprocesoru nebo samostatně, to záleží na konkrétní konstrukci mikrokontroleru.

Některé platformy pro vývoj aplikací pro WSN

Výzkum v oblasti WSN byl započat v 90. letech minulého století. Pro vývoj byly realizovány různé platformy, z nichž se uplatnily např. Mica, Mica 2 - velmi rozšířený typ uzlu, často zmiňovaný v odborných studiích. Dalším příkladem je Imote 1.0, 2.0, který nabízí výkonnější procesor za cenu vyšší spotřeby. Uzly Spec zaujmou malými rozměry v jednotkách milimetrů. Dále existují např. uzly EPIC a Sun SPOT. Poslední zmíněný nabízí možnost programování v Javě a také je energeticky náročnější.

Obrázek 2.1: Příklad mikrokontroleru - Mica 2 [11]

Operační systémy

Operační systémy senzorických sítí jsou také přizpůsobeny požadavkům WSN. Operační systém musí být implementován s ohledem na to, že uzly běžně mívají Flash pamět o velikosti desítek až stovek kilobytů a datovou pamět ve stovkách až tisících kilobytů. Musí tedy být paměťově nenáročný a nechat prostor pro samotné aplikace sítě. Senzorické sítě mají své specifické požadavky na operační systém i z hlediska způsobu řízení událostí, správy procesů, síťování atd. Jedním z typických úkolů je např. řízení aktivního a neaktivního režimu uzlu. Operační systémy používané ve WSN jsou například TinyOS, Contiki, Mantis OS, Nano-RK, LiteOS [10].

2.2 Komunikace uzlů

Komunikační zařízení předává data získaná uzlem základnové stanici. Využívá komunikační protokoly, pracuje s přenosovým médiem, kterým jsou nejčastěji rádiové, ale také ultrazvukové (pro použití pod vodou) a infračervené vlny. Uzly mohou být vybaveny různými moduly pro komunikaci např. přes Bluetooth, BLE (Bluetooth Low Energy), WiFi, ZigBee. Základnové stanice mohou být vybaveny ještě navíc dalšími moduly pro komunikaci se sítí uživatele.

Uzly v bezdrátových sítích fungují s omezenou zásobou energie a komunikace mezi nimi je energeticky nejnáročnější činností. Komunikační protokoly se tedy zaměřují především na optimalizaci využití energie.

Přímá a nepřímá komunikace

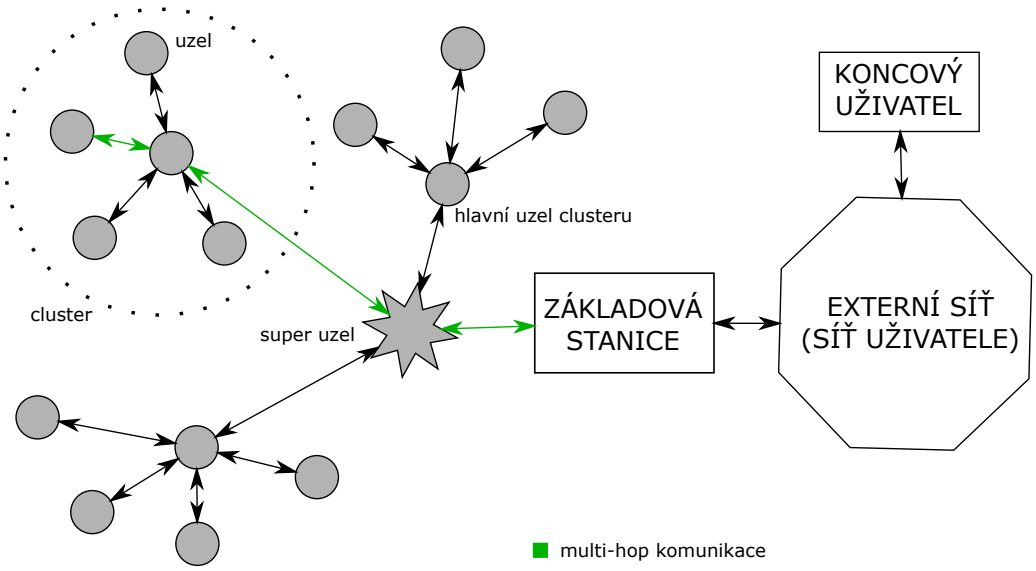
Komunikace může probíhat přímo, zasláním paketu z uzlu do základnové stanice (single hop síť), je proveden jeden přeskok při přenosu. Tento typ komunikace lze uvažovat jen v případě velmi malé sítě, na krátkou vzdálenost, kde nedojde k velkému plýtvání energie při přenosu paketu.

V případě rozlehlých sítí je dávána přednost hiearchickému uspořádání. Síť je rozdělena do tzv. clusterů. V čele jednoho clusteru je hlavní uzel clusteru (cluster-head), který se stará mimo jiné o komunikaci mezi členy clusteru (přímá komunikace) a komunikaci mezi sousedními hlavními uzly clusteru. Výsledkem je přímá nebo nepřímá komunikace mezi hlavním uzlem clusteru a základnovou stanicí. Zde platí podobné omezení, tedy že hlavní uzly clusteru mohou být příliš vzdálené základnové stanici sítě, a tím je přenos neúsporný nebo rovnou nemožný.

Dalším druhem komunikace je nepřímá komunikace pomocí více přeskoků - multi hop. Paket je směrován pomocí více přeskoků mezi uzly, až dorazí do základnové stanice. Pro tento způsob komunikace není nutná topologie clusterů, ale některé uzly sítě musejí navíc umět předávat data. Multi hop sítě umožňují rozšířit její rozsah [14].

Používá se convergecast směrování, to znamená, že se data naměřená uzly postupně směrují do jednoho bodu, např. základnové stanice.

Možné uspořádání WSN je vidět na následujícím obrázku č. 2.2.



Obrázek 2.2: Možné uspořádání WSN

2.3 Požadavky na síť

Síťe mají podle svého konkrétního využití různé požadavky na hardware, topologii, bezpečnost, efektivní napájecí systém, rychlost reakce senzorů na změny měřených veličin, rychlost přenosu v síti a další. Rychlost přenosu je zajištěna také například posíláním krátkých paketů.

Důležité požadavky, které jsou pro WSN společné, jsou následující.

2.3.1 Úspora energie

Důležitý faktor, který ovlivňuje to, jak je síť implementována, je spotřeba energie. Vzhledem k tomu, že komponenty WSN mají často omezené možnosti napájení, bývá zde úspora energie prioritou. Největší spotřebu energie představuje aktivní vysílání a příjem. Je potřeba omezit výkon vysílače a také dobu, kdy je aktivní přijímač. Proto se synchronizuje čas vysílaní s dobou, kdy je přijímač aktivní.

Energii je možné šetřit i tím, že uzel obsahuje hlavní procesor a jemu dva podřízené procesory, které jsou vybavené pouze omezenými instrukcemi a mají minimální spotřebu. Jsou jimi komunikační procesor (vybaven komunikačním modulem, např. BLE) a procesor pro komunikaci se senzory. Hlavní procesor může tedy být neaktivní např. v době snímání dat senzory.

2.3.2 Levná implementace

V bezdrátových senzorických sítích je z důvodu velkého počtu senzorů, kterých můžou být až desetitisíce, také požadavek na jejich nízkou cenu, levná implementace může být ale i důvodem nespolehlivosti jednotlivých komponent. Vývoj je zaměřen na jednočipové systémy, požadavek na nízkou cenu znamená také minimum součástí.

2.3.3 Bezpečnost

V senzorických bezdrátových sítích bývají stěžejní data, nikoliv uzly. Je důležitá sledovaná událost a uzly, které ji sledují, mohou být dokonce redundantní. S ohledem na tento data-centrický přístup a také na typ a citlivost zpracovávaných dat, může být zásadním požadavkem uživatele bezpečnost sítě. K tomu se využívá zabezpečovacích protokolů. Kromě napadení může být funkčnost sítě ohrožena i technickou chybou, jako je výpadek uzlu. Tam, kde by taková chyba mohla mít vážné důsledky (aplikace používané v armádě, zdravotnictví), je snaha omezit chyby pomocí vhodného, odolného hardwaru a algoritmů, které zajistí, aby výpadek komponent neovlivnil fungování sítě jako celku. Obecně ve všech sítích je důležité zachování integrity přenášených dat.

2.3.4 Lokalizace

Lokalizace senzorů je další požadavek senzorických sítí. K efektivnímu zpracování dat je třeba určit polohu, kde byla získána. Jednou z možností je k lokalizaci využít systém GPS. Miniaturní GPS přijímače již dovolují realizovat takové WSN komponenty, ale problémem zůstává jejich spotřeba. Navíc uzly sítě mohou být umístěné i mimo dosah GPS signálu, takže je třeba jiných lokalizačních metod, které určují buď polohu uzlů vůči bodům, které stanoví vysílače polohy, nebo relativně k bodu, kde se nachází vstup do sítě.

2.3.5 Časová synchronizace

Pro senzorické sítě je důležitá časová synchronizace uzlů a informace o čase, kdy byla data získána. Je nutná pro koordinovanou činnost uzlů a také pro úsporu energie. Časová synchronizace je stěžejním tématem této práce a bude detailně popsána v samostatné kapitole.

2.3.6 Topologie a škálovatelnost

Zásadní faktor designu WSN je efektivní rozmístění jejích částí. Topologicky správně navržená síť by měla být energeticky méně náročná, odolnější vůči výpadkům senzorů a také proti rušení při přenosech. Topologie sítě se průběžně dočasně mění tím, jak jsou uzly aktivní nebo spící. V senzorické síti by mělo být možné provádět rekonfiguraci i za chodu sítě, například přidáním uzlu, ne pouze při jejím počátečním nastavení. Pokud je možné provádět takto dodatečné úpravy, říkáme, že je síť škálovatelná. Existuje mnoho typů a dělení topologií, v následující části budou popsány pouze některé základní.

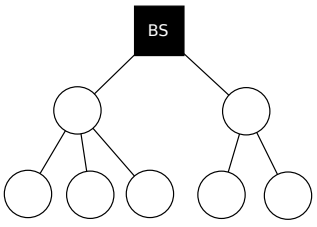
Typy topologií

Topologii je možné rozdělit na základní dva typy - jednoúrovňová a hiearchická topologie.

Hiearchická topologie

V hiearchické strukutuře jsou uzly různé úrovně v různých vrstvách a nemusí mít stejnou funkci. Např. v clusterové topologii existují základní snímající uzly a hlavní uzly clusteru, které komunikují se základnovou stanicí a agregují data ze snímajících uzlů. Je možné vnořováním tvořit další clustery v síti. Clusterová topologie je znázorněna na obr. 2.2 na straně 12.

Ve stromovém uspořádání (obr. 2.3) komunikuje uzel určité úrovně s uzly nižší a vyšší úrovně ve vzdálenosti jednoho přeskoku. Buduje se hiearchická strukutra, aby bylo možné předávát pakety od uzlu s nejvyšší úrovní hiearchie směrem k nižším uzlům - od kořenu k listům grafu. Kořenový uzel je zde označen jako BS (base station, základnová stanice).

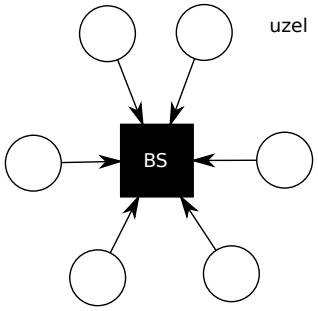


Obrázek 2.3: Stromová topologie

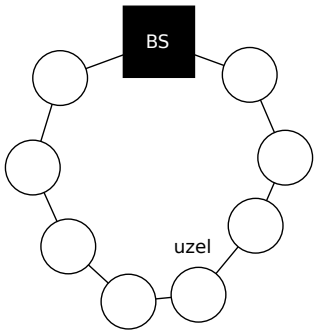
Jednoúrovňová topologie

V jednoúrovňové topologii spolu uzly komunikují na stejné úrovni, v různém logické uspořádání, kterým může být např. kruh, hvězda nebo jiné uspořádání, které nemá určitý geometrický tvar, tzv. smíšená topologie.

Logické uspořádání do hvězdy je nejjednodušším modelem uspořádání. (obr. 2.4) Uprostřed se nachází základnová stanice, která je propojena s okolními uzly na vzdálenost jednoho přeskoku. Základnová stanice zde plní funkci serveru a přidružené uzly jsou klienti. Efektivita tohoto typu uspořádání klesá s velikostí sítě, je vhodná pouze pro velmi malé sítě.



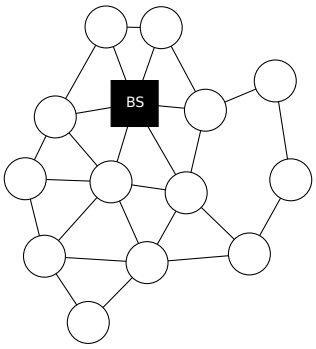
Obrázek 2.4: Hvězdicová topologie



Obrázek 2.5: Kruhová topologie

Další jednoúrovňovou topologií je uspořádání do logického kruhu. (obr. 2.5) Každý uzel má právě dva sousedy a komunikace probíhá v jednom směru. Může probíhat i v obou směrech, je tak možné zálohovat i pro případ

výpadku uzlu. Co se týče použití, je to podobné jako u hvězdicové topologie. Efektivita je zajištěna pouze při malém počtu uzlů, při větším by bylo směrování příliš energeticky náročné.



Obrázek 2.6: Smíšená topologie

U smíšené topologie (obr. 2.6) se uplatňuje multi hop směrování. Používají se různé algoritmy, jak si zpráva hledá cestu v síti, např. hledání kostry grafu. Je to důležité pro efektivní směrování a úsporu energie [3].

2.4 Architektura sítě

Protokolový zásobník WSN sítí se skládá z fyzické, linkové, síťové, transportní a aplikační vrstvy. Napříč těmito vrstvami fungují ještě dodatečné vrstvy - tzv. planes.

Vrstva správy úloh (task management plane) zajišťuje rozdělení úloh pro jednotlivé senzory tak, aby běh sítě byl optimální z hlediska úspory energie. Takže např. v přiděleném úseku sledují objekt zkoumání sítě v určitém čase jen ty senzory, které mají dostatek energie.

Vrstva správy pohybu (mobility management plane) monitoruje pohyb uzlů a umožňuje jim sledovat změny lokace jejich sousedů.

Vrstva správy energie (power management plane) řídí využívání energie uzly, např. vypnutí přijímače uzlu po obdržení zprávy, předávání informace mezi uzly o stavu jejich energie.

2.4.1 Fyzická vrstva

Fyzická vrstva je rozhraní, které umožňuje přenos dat skrze komunikační médium. Senzorické sítě využívají často rádiové vysílání, ale také ultrazvuk, světlo atd. Rádiové vysílání probíhá na ISM pásmech. Číslicový signál je modulovaný na analogový frekvenčně nebo fázově. Podle typu modulace je přenášeným hodnotám přiřazena různá frekvence či fáze. Protože ke stejnému kanálu přistupuje více účastníků přenosu, využívá se časový, frekvenční a kódový multiplex. V případě časového multiplexu (TDMA) se používají časová okénka, každý vysílač má své, takže se neruší a mohou vysílat na stejné frekvenci. Frekvenční multiplexování (FDMA) znamená souběžné vysílaní na různých frekvencích. Také je možné multiplex realizovat kódováním (CDMA), zde sdílí vysílače frekvenční pásmo po celou dobu vysílání a jejich data jsou kódována a každý vysílač má svou ortogonální čipovou sekvenci, ze které příslušný přijímač odvodí konkrétní data i přes přítomné vysílání ostatních. K realizaci přenosu v senzorických sítích se nehodí WiFi standard IEEE 802.11, kvůli požadavkům na omezenou spotřebu energie. Využívají se standardy z řady IEEE 802.15 [12], např. 802.15.1 (Bluetooth, BLE) nebo 802.15.4 (LR-WPAN).

2.4.2 Linková vrstva

Tato vrstva zajišťuje přenos dat mezi sousedními uzly. Jednotkou přenášené informace je rámec. Linková vrstva má za úkol zajistit fuknce jako je synchronizace rámců, využití přenosového pásma, řízení toku dat a zajištění

integrity rámce - řízení chyb. K tomu se používá ve WSN kontrolní součet CRC. Porušení integrity rámce se oznamuje kladným či záporným potvrzováním, v závislosti na použitém protokolu. Příkladem linkového protokolu pro WSN je IEEE 802.15.4 [12].

2.4.3 Síťová vrstva

Hlavním úkolem síťové vrstvy je směrování paketů napříč sítí. Pricipy směrování v bezdrátových senzorických sítích jsou odlišné od směrování například u TCP/IP protokolů, proto se používají specifické algoritmy, jako je zaplavování (flooding), řízené šíření paketu (directed diffusion), geografické směrování a další. Také se dá směrování rozdělit na směrování podle požadavku a předem určené podle směrovací tabulky.

Zaplavování je jednoduchý mechanismus, jak rozšířit po síti informaci o nějaké nastalé události. Jeden z uzlů v oblasti výskytu události začne zaplavovat síť zprávou. Lze nastavit rádius dosahu, kam až se má informace dostat. Jeho velikost je dána počtem přeskoků. Nevýhodou tohoto směrovacího algoritmu je samozřejmě redundance zpráv, které zatěžují síť. Informace dostávají i uzly, které je nepotřebují.

Optimalizací předchozího algoritmu je řízené šíření. Základnová stanice chce získat nějakou informaci, a tak odešle požadavek do sítě a uzly mající tuto informaci ji odpoví.

Geografické směrování využívá polohy uzlu, je tedy použitelné, pokud uzly svou polohu znají. Předávají paket sousedovi, který je nejblíže cílovému místu. Dále je možné šířit paket při výběru více směrů stále jedním směrem, což vede k neoptimální, ale jisté cestě.

2.4.4 Transportní vrstva

Transportní vrstva slouží k řízení toku dat mezi koncovými aplikacemi, ochraně proti zahlcení a zajišťuje detekci chyby a obnovení přenosu po chybě. Použití TCP a UDP protokolu je u WSN nevhodné. Protokol UDP poskytuje nezabezpečený přenos dat. Nehodí se pro potřeby senzorických sítí, protože nemá zabudované řízení toku dat a ochranu proti zahlcení. Ztráta paketu způsobí opakování přenosu, a tím zvýší spotřebu energie. TCP protokol také není vhodný pro velkou režii s navazováním spojení a svou impelementací generuje z pohledu WSN značná zpoždění. Algoritmy použité ve WSN na transportní vrstvě jsou opět přizpůsobené požadavkům těchto sítí. Délka paketu v senzorických sítích je z úsporných důvodů omezená, přenosový kanál bývá znatelně zatížen rušením, přenos přes velký počet uzlů vede

k velkým zpožděním. Je třeba minimalizovat počet opakování přenosu.

S přenosem na transportní vrstvě souvisejí pojmy unicast a broadcast. Unicast je přenos mezi dvěma uzly sítě. Broadcast je všesměrové vysílání šířené do všech uzlů.

Protokoly transportní vrstvy jsou dvojího typu - upstream a downstream. Upstream je typ přenosu ze senzorů do základnové stanice, shromažďují se naměřená data. Downstream směřuje naopak ze základnové stanice k senzorům, je jím např. broadcast synchronizační zprávy [12].

2.4.5 Aplikační vrstva

Hlavní rolí aplikační vrstvy je poskytnout abstrakci fyzických komponent sítě aplikacím. Zajišťuje konverzi aplikačních dat na data, která jsou vhodná pro přenos. Jednou z funkcí aplikační vrstvy je komprese dat, která může být ztrátová. Ta způsobí značnou redukci velikosti přenášeného paketu, nebo bezeztrátová, která data neporuší. Pro senzorické sítě mají naměřená data zásadní význam, proto se používá bezeztrátová komprese. Protokolem bezeztrátové komprese je např. Sensor LZW, který funguje na slovníkové kompresi, nazhrazuje řetězec dat symbolem ze slovníku.

Další funkcí je zpracování dotazu uživatele, aby mohl řídit senzorickou síť. Jsou dotazy typu push, kdy uzly poskytují naměřená data vyšším vrstvám hierarchie. Zpracování dotazu pull metodou znamená dotaz od základnové stanice směrem k senzorům. Příklady protokolů zpracování dotazů jsou např. COUGAR a SQTL [5].

Úlohou aplikační vrstvy je také monitorování sítě při nastalé události nebo po celou dobu běhu. Podle potřeby frekvence monitorování se liší použité algoritmy. Příkladem monitorovacího protokolu je protokol MANNA [5].

3 Úvod do časové synchronizace

Časová synchronizace je ve WSN potřebná z mnoha důvodů. Senzory bezdrátové sítě obecně potřebují vykonávat synchronizovanou činnost, ale nelze jednoduše stanovit, jaká metoda časové synchronizace je nejlepší. WSN aplikace mají podle své funkce různé požadavky, zda bude synchronizace probíhat podle logického nebo reálného času, pouze uvnitř systému, nebo vzhledem k vnějšímu médiu. Naměřená data by ve většině případů postrádala význam bez lokalizace a časového určení - kdy a kde byla získána. Je třeba navrhnout časovou synchronizaci se snahou eliminovat chyby vzniklé odchylkami času a také s ohledem na úsporu energie, která je jednou z priorit při provozu WSN [5].

3.1 Pojmy - referenční čas

Referenční čas je požadovaný čas, ke kterému synchronizujeme bezdrátovou síť nebo její jednotky. Uvnitř sítě si často senzory vystačí s logickým časem a svými hodinami.

Globální čas (physical time)

Tímto časem se myslí reálný čas, který je obecně platný podle domluveného schématu. Tím je například UTC, koordinovaný světový čas, který je založený na principu atomových (cesiových) hodin. Jedna sekunda tohoto času je definována jako určitý počet kmitů atomů cesia. UTC je systém měření času navazující na GMT, který je založený na rotaci Země. Mezi těmito časovými standardy existují minimální odchylky, které jsou vyrovnávány přidáváním přestupných sekund.

Logický čas

Logický čas se používá ve chvíli, kdy je třeba časově určit sled událostí, jak nastaly. Nekoreluje s globálním časem, ale platí uvnitř systému. K časovému určení se používá časová značka (time stamp). Většinou stačí např. k zajištění bezchybné výměny zpráv.

3.2 Synchronizační chyby

Synchronizačním chybám obecně nelze nijak preventivně zabránit, řešením je tedy opravování vznikajících odchylek. Z hlediska efektivity by vždy bylo nejvhodnější korigovat frekvenci hodin, ta však závisí i na vnějších podmínkách, které do jisté míry nelze ovlivnit, zvláště v hodinách umístěných v terénu. Nejčastěji tedy korigujeme vzniklou odchylku tím, že opakovaně měníme nastavení hodin. Existují velmi přesné zdroje času, jako jsou atomové hodiny s minimálními odchylkami, běžné hodiny potřebují pravidelně seřizovat na správný čas. Přesnost hodin je dána také kvalitou a konstrukcí hodin, tedy u levných implementací k odchylkách dochází ve větší míře.

Hodiny mají tendenci se předcházet nebo zpožďovat. Zpoždění je snazší na opravu, u předcházejících se hodin je problém v tom, že nelze vracet čas zpět. Když jdou hodiny napřed, je možné přeskočit určitý počet tiků, a tím dosáhnout korekce. Tu je lepší provádět postupně, aby nedošlo k velkému výkyvu času.

3.2.1 Frekvenční odchylka

Clock skew

Skew je rozdíl frekvence hodin oproti frekvenci perfektních hodin. Frekvence je dána jako počet tiků hodin za sekundu reálného času. V souvislosti s termínem skew se myslí téměř výhradně relativní skew, tedy rozdíl frekvencí dvou vybraných hodin. Na obrázku 3.1 je tato odchylka značena modře. Modré hodiny jsou rychlejší oproti černě znázorněným perfektním hodinám [7][6].

Clock drift

Tato odchylka je důsledkem změny frekvence kmitání oscilátoru v hodinách uzlu. Je nekonstantní a do značné míry závislá na vnějších podmínkách, zejména na teplotě. Drift způsobuje, že i po synchronizaci se časem hodiny rozejdou. Teplotní rozdíl 10 stupňů Celsia může například způsobit odchylku frekvence 2 minuty za rok, při rozdílu 20 stupňů už je to 8 minut ročně. [7] Čas ukazovaný na hodinách nabývá v různých chvílích kladných a záporných odchylek vůči správnému času. Na obr. 3.1 je toto znázorněno červenou křivkou.

Stabilita krystalového oscilátoru bývá okolo +/- 20 ppm (parts-per-milion), což je počet kmitů, který se odchýlí od jednoho milionu kmitů. Existuje

mnoho druhů oscilátorů, které mají různou stabilitu, často používaným typem je rubidiový oscilátor. Za jeden z nejstabilnějších je považován již zmíněný cesiový oscilátor (atomové hodiny) [2].

Drift je možné vyjádřit podle následujícího vztahu:

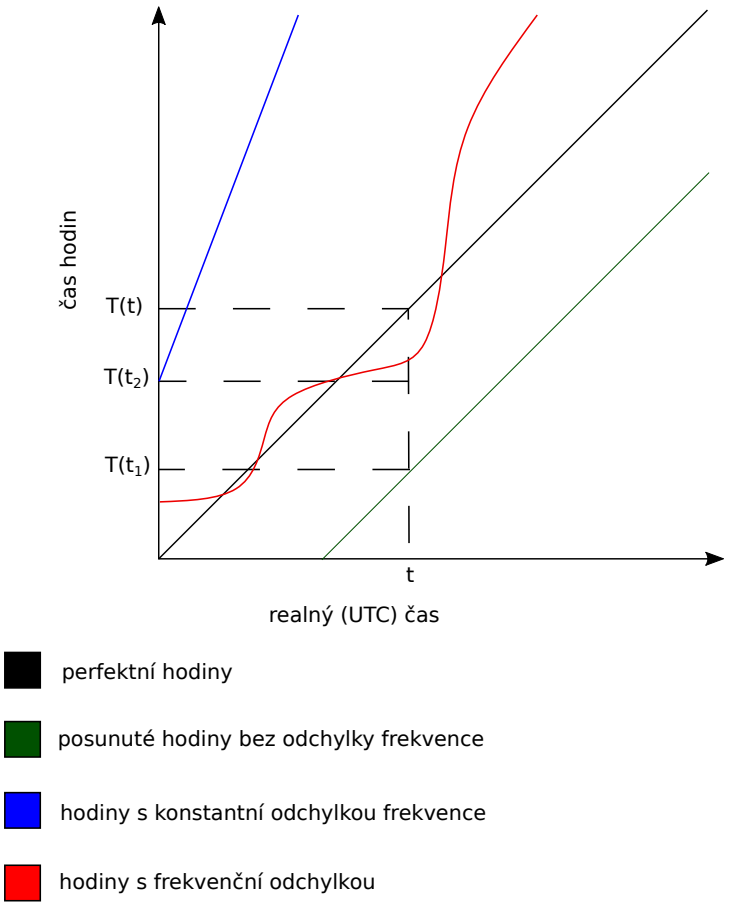
$$1 - \rho \leq \frac{dC(t)}{d(t)} \leq 1 + \rho$$

kde C(t) ... čas hodin, t ... reálný (UTC) čas, ρ... maximální odchylka.

3.2.2 Časová odchylka

Clock offset – posuv hodin

Je rozdílem hodin uzlu oproti správnému času (UTC). Relativní offset je rozdíl hodin dvou uzlů. Samotný offset je znázorněn zeleně na obr. 3.1.



Obrázek 3.1: Srovnání časových odchylek

Aby hodiny plnily správně svou funkci, je třeba je synchronizovat, to znamená v ideálním případě kompenzovat všechny typy odchylek, které se u nich vyskytují. Někdy je to pouze jeden druh odchylky, např. frekvenční. Často ale nesoulad hodin způsobuje kombinace všech odchylek.

U hodin, které potřebují jen seřizovat a nejsou součástí sítě, kterou je potřeba synchronizovat jako celek, je situace jednodušší, většinou stačí je pravidelně seřizovat vůči UTC času. K tomu je možné použít atomové hodiny, GPS, synchronizační pulzy vysílané rádiem atd. Kompenzujeme tedy offset. Ideálně systém kombinuje atomové hodiny pro korekce drobnějších ochylek a GPS pro opravu dlouhodobých odchylek [7].

V senzorických sítích je potřeba minimalizovat odchylku mezi uzly. Některé synchronizační protokoly pro WSN se zaměřují pouze na korekci offsetu, jiné používají algortimy i k opravě frekvenční odchylky, takovým je např. použití lineární regrese.

3.3 Základní dělení metod časové synchronizace

Mnoho protokolů pro časovou synchronizaci uzlů je založených na GPS, nicméně tyto metody selhávají u WSN z toho důvodu, že senzory mohou být umístěny mimo GPS signál, např. pokud se nacházejí pod vodou a jiných těžce dostupných místech, nebo v budovách. Další nevýhodou GPS je pro WSN energetická náročnost. Přijímač pro GPS signál je pro potřeby uzlu WSN sítě (levné zpracování, malé provedení atp.) příliš zatěžující komponentou. Omezený hardware uzlů znamená i nepřesné hodiny, proto je snaha o korekci času mezi uzly. To je hlavním důvodem pro vývoj softwarově orientovaných metod synchronizace ve WSN.

3.3.1 Požadavky na časovou synchronizaci

Základním požadavkem na jakýkoliv synchronizační mechanismus je vyrovnat se s omezeními provozu bezdrátových senzorových sítí. Požadavky a omezení provozu nejsou vždy v souladu, např. chceme maximalizovat přesnost synchronizace a na druhou stranu minimalizovat spotřebu energie senzorů.

Z hlediska algoritmů pro úsporu je synchronizace důležitá. Uzly mohou šetřit svou energii tím, že v nečinnosti přejdou do úsporného režimu, tím, že na čas vypnou snímací senzor nebo přijímač. Takto spící uzel se zaktivuje v případě potřeby. Pro koordinaci spánku a aktivity uzlu je potřeba nača-

sovat tyto akce a s přesnější synchronizací lze dosáhnout větší úspory, minimalizuje se čas, kdy je uzel zbytečně plně aktivní. Rozdíl v časování je popsán dále
v části o pre-facto a post-facto synchronizaci.

Přesnost synchronizace je dalším faktorem, přesné metody, jakými je synchronizace pomocí GPS a NTP protokolu, nejsou vzhledem k vlastnostem bezdrátových sítí praktické. Pro sítě např. bývá důležité zachovat jejich dynamickou topologii a nebo je GPS pro počet senzorů příliš finančně náročné řešení.

Bezpečnost časové synchronizace může být narušena mnoha typy útoků, nejčastěji DoS útoky a útoky na komunikační kanál. Výměnu zpráv mezi uzly může chtít někdo narušit, řešením je zabezpečit jejich komunikační protokoly.

3.3.2 Absolutní a relativní synchronizace

Jako absolutní je brána sychronizace k globálnímu času, který byl popsán v předchozí kapitole jako všeobecně uznávané časové schéma.

Aplikace, které potřebují synchronizaci podle globálního času jsou obecně takové, které sbírají a vyhodnocují reálná data, např. bezpečnostní aplikace pro zaznamenání pohybu osob, reálný čas je také nutnost při práci s rychlostí, zrychlením atd. Příkladem je síť, kde senzory snímají polohu sledovaného objektu vůči sobě. Další informace o tomto objektu vzniknou spojením dat (data fusion) z více senzorů, právě např. rychlost, ale i tvar objektu. Právě při fúzi dat hraje reálný čas důležitou roli.

Je potřeba nejen ke správnému fungování, ale i k ovládání sítě ze strany uživatele, s čímž se můžeme běžně setkat v domácích chytrých systémech, kterým zadává člověk nebo vnější systém časově specifické úkoly, např. fungování v nočním režimu apod.

Relativní synchronizace je synchronizace hodin uvnitř systému vůči sobě. Hodiny jdou stejně podle nějakého zdroje hodinového signálu, který ale nekoreluje s globálním časem. Tyto údaje nejsou pak navenek porovnatelné, ale stačí pro fungování daného systému.

3.3.3 Pre-facto a post-facto synchronizace

Pre-facto synchronizace

Jedná se o případ, kdy je potřeba synchronizovat snímaní senzoru pro předpokládaný časový okamžik (time-triggered event). Mimo tuto dobu není potřeba synchronizace.

Post-facto synchronizace

Post-facto synchronizace znamená, že uzly fungují se svými vlastními hodinami nesynchronizovaně a synchronizace proběhne až v okamžiku, kdy je potřebná. To probíhá na základě události při snímání senzoru (event-triggered synchronization). Uzly si zaznamenají čas výskytu události v závislosti na svých hodinách. Následně jiný uzel vyšle do broadcastu pomocí rádia synchronizační puls, který uzly berou jako okamžitou časovou informaci a na základě ní mohou určit svou odchylku v době události. Princip post-facto synchronizace vedl ke vzniku RBS protokolu [17].

Post-facto synchronizace funguje v protokolech, kdy synchronizace proběhne na žádost uzlu, jinak by byla příliš energeticky náročná, kvůli častému vyrovnávání odchylek. Jinak je omezena dosahem vysílače, který vysílá synchronizační puls. Je využitelná v řadě lokalizačních systémů, kdy stačí porovnat čas synchronizačního signálu s časem na množině snímajících senzorů.

3.3.4 Sender-receiver / receiver-receiver

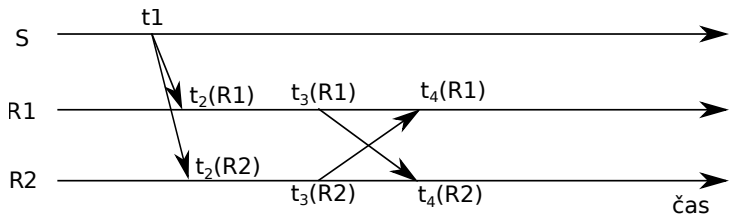
Sender-receiver

V protokolech založených na sender-receiver principu si přijímač upraví čas na čas vysílače. Pomocí časových značek v paketech, které vysílač periodicky vysílá, určí offset vůči vysílači. Tento princip je možný nejen pro párovou synchronizaci, ale i pro synchronizaci v rámci sítě. Jeden uzel se v tomto případě stane referenčním, se kterým se ostatní synchronizují. Přímý soused referenčního uzlu si podle něj upraví hodiny a stane se referenčím uzlem pro svého dalšího souseda. Synchronizační mechanismus by se tak dal znázornit stromovou strukturou. Tento postup není příliš vhodný z důvodu snadného rozšíření případné chyby. Vhodnější varitantou je metoda, kdy každý uzel zná identitu referečních uzlů a směruje k nim žádost o synchronizaci.

Příkladem synchronizace na principu sender-receiver jsou NTP (Network Time Protocol) [1], LTS (Lightweight Time Synchronization), TPSN (Timing-Sync Protocol for Senzor Nets) [13], HRTS (Hiearchy Referencing Time Synchronization).

Receiver-receiver

V tomto typu protokolů se příjemci synchronizují mezi sebou, ne s odesílatelem synchronizačního paketu. Je zde opět referenční uzel, který vysílá paket. Není třeba, aby obsahoval časovou značku, protože není důležité, kdy byl odeslán, pouze kdy byl přijat příjemci. Předpoklad je, že dva příjemci přijmou stejný paket v přibližně stejný čas (t2). Vzájemně si pošlou zprávu s časem, kdy paket přijali (t3), a na základě toho vypočítají offset vůči sobě.



Obrázek 3.2: Průběh výměny zpráv při receiver-receiver synchronizaci

Je teoreticky možné synchronizovat velké množství uzlů pomocí jednoho synchronizačního paketu, ale platí, že s vyšším počtem synchronizačních paketů roste přesnost synchronizace, nevzniká zpoždení při příjmu zprávy mnoha uzly.

Příkladem receiver-receiver protokolu je RBS (Reference Broadcast Synchronization) [8].

3.3.5 Tradiční způsob synchronizace (TTS)

Tradiční časová synchronizace funguje na principu sender-receiver. Odesílatel periodicky posílá zprávu s časem svých hodin jako časovou značku příjemci. Např. rádiové stanice mohou pravidelně posílat signály s přesností na jednu mikrosekundu. Tato metoda pak může poskytnout synchronizaci příjemci s možnou odchylkou okolo deseti mikrosekund.

3.3.6 NTP protokol

Pro systém, který má možnost přístupu k internetu, je jednoduchou variantou časové synchronizace NTP protokol. Je široce použitelný, jeho výhodami jsou robustnost vůči chybám, samo-konfigurační vlastnosti a škálovatelnost. Z důvodů uvedených v předchozím odstavci ale není vhodným schématem pro WSN, ačkoliv spojením GPS a NTP lze dosáhnout přesné synchronizace v řádu mikrosekund. Dalšími komplikacemi při použití ve WSN jsou omezená šířka pásma, dynamická topologie sítě atd. NTP je síťový

UDP/IP protokol. Použití UDP je nespolehlivé tím, že nezaručuje doručení dat. Je ale vhodným nástrojem pro NTP, protože spolehlivý TCP protokol by při ztrátě zprávy posílal znovu tuto zprávu, ale její časová informace by byla kvůli zpoždění neplatná.

U NTP funguje externí synchronizace, referenčním časem je UTC. Protokol funguje na principu sender-receiver. NTP klient, který chce synchronizovat své hodiny, k tomu využívá časové servery, synchronizované s externími hodinami, např. GPS nebo jiné zdroje jako radiové nebo satelitní systémy. Současná verze protokolu je NTP 4. Jeho možná odchylka má být v řádu mikrosekund pro sítě LAN a jednotky milisekund v sítích WAN [1].

Časové servery jsou v hiearchické stromové struktuře vrstev (strata). Primární server v nejvyšší vrstvě je napojen k externímu zdroji přesného času, kterým je UTC čas. Dále hiearchie pokračuje se sekundárními servery a klienty. S rostoucími vrstvami klesá přesnost hodin, ale výhodou tohoto schématu je, že nedochází k přetížení primárního serveru. Vrstev je v tomto protokolu nejvýše patnáct.

NTPv4 démon je uživatelský proces chráněný šifrováním, kvůli potenciálnímu riziku neoprávněné změny systémového času. Existuje zjednodušená verze SNTP (Simple Network Time Protocol), kde není nástroj k tomu, aby klient s podporou SNTP mohl sloužit jako server pro další klienty. SNTP si nepamatuje předchozí komunikaci, neuvažuje zpoždění paketů. Používá se tam, kde není potřeba nejpřesnější synchronizace.

Protokol může fungovat ve třech režimech provozu: server-klient, broadcast a symetrický režim. Server-klient funguje na pull metodě, broadcast používá push – server „tlačí“ synchronizaci klientům.

4 Synchronizační protokoly bezdrátových senzorických sítí

Protokoly pro časovou synchronizaci také vycházejí z obecných schémat časové synchronizace, která byla popsána, najdeme mezi nimi množství protokolů založených na principu sender-receiver (TPSN, FTSP, DMTS) i receiver-receiver (RBS, E-RBS). Protokol RTSP lze lépe popsat jako klient-server strukturu, protože zprávy zde proudí v obou směrech - od odesílatele k příjemci a naopak.

Vývoj v této oblasti přinesl různé nové algoritmy, jak kompenzovat časové odchylky. Některé z protokolů kompenzují jak frekvenční (skew), tak časovou odchylku (offset). Některé se vyrovnají pouze s offsetem. V následující části je popsáno šest protokolů, které byly vybrány jako zástupci nejznámějších a bežně využívaných synchronizačních protokolů ve WSN.

4.1 TPSN (Timing-sync Protocol for Sensor Networks)

TPSN protokol je synchronizační protokol založený na principu sender-receiver. Synchronizace pomocí TPSN probíhá ve dvou fázích, v první fázi se vytváří hiearchická topologie sítě, kdy je uzlům přiřazena úroveň v závislosti na jejich postavení v síti, přičemž uzlu na vrcholu hiearchie je přiřazena úroveň nula. Ve druhé fázi, která se označuje jako synchronizační, se synchronizují uzly úrovně i s uzly na úrovni i-1. Výsledkem jsou uzly synchronizované s časem nejvyššího uzlu.

Předmětem synchronizace je určitý počet uzlů, rozprostřených v nějaké oblasti. Každý uzel eviduje čas pouze jako hodnotu svého jednoho registru v CPU, který se inkrementuje počtem tiků krystalového oscilátoru. Jiný pojem o čase nemá.

4.1.1 Fáze přidělení úrovně

Prvním krokem v této fázi je přidělení úrovně *0* hiearchicky nejvyššímu uzlu. Tím je označen jako kořenový uzel - *root*. Kořenový uzel vyšle všem uzlům sítě paket, který obsahuje jeho identitu a úroveň v hiearchii. Sousední uzly přijmou tento paket, a na základě informací v něm si samy přiřadí úroveň o jedna větší. Následně stejně jako kořenový uzel vyšlou paket, podle kterého si přiřadí úroveň další uzly zvětšením o jedna, a zároveň, pokud již znají svou úroveň, odmítnou stejný paket.

Přidělení úrovní lze také dosáhnout algoritmicky složitější metodou minimální kostry grafu.

4.1.2 Synchronizační fáze

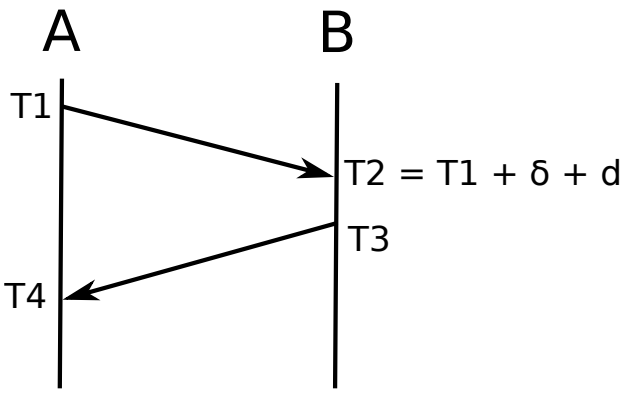
V synchronizační fázi dochází k synchronizaci mezi dvěma uzly, např. A a B. Pokud si síť představíme jako graf, k synchronizaci dochází postupně po všech hranách mezi jednotlivými uzly směrem od uzlu hiearchicky nejvyššího. Využívá se sender-receiver synchronizace.

Postup synchronizace je následující: v čase *T1* vyšle uzel A synchronizační puls. Ten je přijat uzlem B v čase *T2*. Rozdíl časů *T1* a *T2* je součet odchylky hodin uzlů A a B a zpoždení při přenosu pulsu. V čase *T3* uzel B pošle do A potvrzovací paket s časy *T1, T2, T3* a úrovní uzlu B. Tento paket je přijat uzlem A v čase *T4*. Za předpokladu, že během jejich komunikace nedošlo ke změně odchylky hodin a zpoždění při přenosu paketu, uzel A může zjistit odchylku δ a zpoždění d:

$$\delta = \frac{(T2-T1)-(T4-T3)}{2} \qquad d = \frac{(T2-T1)+(T4-T3)}{2}$$

Na základě znalosti odchylky si poté uzel A nastaví své hodiny na čas hodin uzlu B [13].

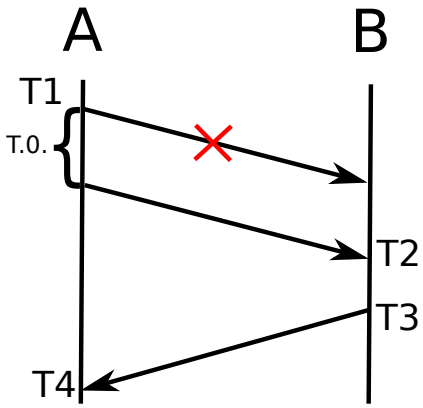
Synchronizační fázi zahájí kořenový uzel vysláním inicializačního paketu. Uzly, které jej přijmou, zahájí výměnu zpráv a synchronizují časy svých hodin na čas hodin kořenového uzlu. Při synchronizaci hodin dvou uzlů s úrovní i uzly úrovně $i+1$ čekají náhodný časový interval, než začnou vysílat synchronizační pulsy. Tím je zajištěno, že začnou svou synchronizační fázi až poté, co ji dokončí uzly i-té úrovně, a synchronizují se tak ke správnému času. Bezchybná výměna zpráv je znázorněna na následujícím obrázku č. 4.1.



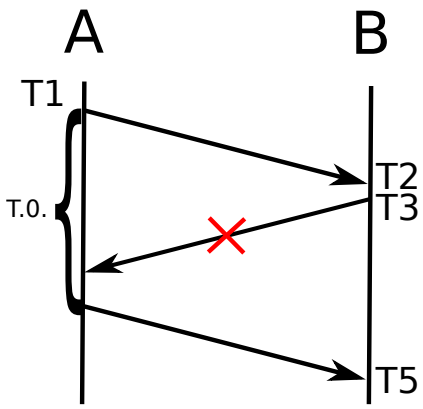
Obrázek 4.1: Bezchybná výměna zpráv

4.1.3 Řešení chybových stavů

Během procesu synchronizace může dojít ke ztrátě paketu či potvrzení. Řešení je, že uzel, který čeká na potvrzení, po vypršení nějaké doby (na obrázku T.O. jako timeout) svůj paket zasílá znovu, dokud mu nepřijde potvrzení. Možné scénaře chyby při výměně zprávy jsou znázorněny na následujících obrázcích č. 4.2 a 4.3.



Obrázek 4.2: Ztráta zprávy



Obrázek 4.3: Ztráta potvrzení

Pokud uzlu sítě chybí úroveň hiearchie, např. byl zapojen nově náhradou za jiný uzel, zjistí svou úroveň vysláním paketu do broadcastu. Obdrží různé hodnoty od sousedních uzlů a sám si přiřadí úroveň o jedna větší, než je nejmenší úroveň sousedního uzlu.

Může se také stát, že uzel B není v provozu, a tak opakovaně nereaguje na žádost o synchronizaci. Znamená to, že uzel úrovně o jedna větší od něj nedostane potvrzení na zasílané synchronizační pulsy. Po určitém počtu pokusů uzel vyšle do broadcastu zprávu pro zjištění úrovně a podle odpovědi některého sousedního uzlu si přiřadí novou úroveň. Stane-li se, že vypadne kořenový uzel, proběhne algoritmus nalezení nového kořenového uzlu, poté následuje znovu fáze přidělení úrovně.

4.2 FTSP (Flooding Time Synchronization Protocol) protokol

Účelem použití FTSP protokolu je časová synchronizace v síti čítající stovky uzlů, s časovou odchylkou maximálně v řádu mikrosekund a robustní síť vůči změnám topologie a výpadkům uzlů. FTSP je algoritmus typu sender-receiver. Podobně jako u TPSN uzly synchronizují čas svých hodin s kořenovým uzlem.

Kořenový uzel pomocí rádia vyšle synchronizační zprávu všem uzlům. Ta obsahuje časovou značku, kterou je globální čas v době odeslání. Příjemci na základě času, kdy obdrží tuto zprávu, mohou určit posuv hodin (offset)

vůči odesílateli. Časové značky příjemců i odesílatele jsou uděleny na MAC vrstvě v čase přenosu. Takto je eliminován posuv hodin, ale zůstává zde problém frekvenční odchylky různých hodin. Tu FTSP protokol řeší pomocí lineární regrese, která představuje aproximaci hodnot časů přímkou metodou nejmenších čtverců. Vyšší přesnosti a také bezpečnosti je zaručeno tím, že se uzel synchronizuje podle časových údajů od více uzlů než pouze jednoho.

4.2.1 Postup synchronizace pomocí FTSP

Bezdrátové senzorické sítě typicky zabírají větší plochu, než je rozsah signálu jednoho uzlu, proto FTSP protokol provádí synchronizaci s použitím multi hop přeskoků. Kořenovým uzlem je vždy jeden vybraný uzel a jejich výběr se střídá. Má za úkol udržovat globální čas a poskytovat synchronizaci uzlům nižší úrovně. Uzly, jak se synchronizují, postupně tvoří strukturu kostry grafu. Tím není třeba jim explicitně předtím vytvářet uspořádání.

4.2.2 Struktura synchronizační zprávy

Zpráva, která je vyslána nejprve kořenovým uzlem, je tvořena preambulí, SYNC byty, deskriptorem zprávy, samotnými daty a končí kontrolními CRC byty. Během přenosu preambule se příjemce naladí na frekvenci příchozího signálu. Ze synchronizačních bytů dokáže určit bitový posuv, který potřebuje ke správnému zpracování zprávy. Deskriptor dále obsahuje další informace, např. velikost dat. CRC součet na konci je kontrolou, že zpráva přišla v pořádku. Informace o čase se přidává za SYNC byty při každém přenosu či příjmu zprávy.

4.3 RBS (Reference Broadcast Synchronization)

RBS protokol je jednoduchý synchronizační protokol typu receiver-receiver. Na začátku synchronizace je vyslán do broadcastu tzv. beacon, což je rámec, který neobsahuje žádnou časovou značku a pro synchronizaci uzlů mezi sebou nezáleží příliš ani na čase, kdy je vyslán. Samozřejmě ale, čím přesnější je tento čas, tím lépe je síť synchronizovaná i ke globálnímu času. Nicméně tento algoritmus byl vyvinut hlavně pro synchronizaci uzlů uvnitř sítě mezi sebou a za účelem odstranit nedeterminismus vysílače. Čas vyslání referenční zprávy neovliví celkovou synchronizaci, protože záleží pouze na čase jejího přijetí. Beacon tedy vysílá k tomu určené zařízení. Zachytí ho sousední uzly,

které si zaznamenají čas přijetí. Poté se synchronizují mezi sebou pomocí zpráv, ve kterých si tento čas vyměňují. Tím zjistí offset vůči sobě a je možná synchronizace [8].

Algoritmus je vhodný zejména pro menší sítě, kvůli množství zasílaných zpráv mezi uzly. Nejmenší možná síť pro použití RBS jsou tři zařízení, jedno vysílající sychronizační rámec a dvě, které se synchronizují.

Rozšíření sítě na více účastníků vyžaduje i vyslání více synchronizačních rámců. To zároveň zvyšuje přesnost synchronizace.

4.3.1 E-RBS (Efficient RBS)

Existuje úprava klasického RBS algoritmu, která omezuje množství posílaných zpráv. U RBS algoritmu všechny uzly posílají svůj čas ve svém dosahu ostatním, za účelem synchronizace. Úspora u E-RBS spočívá v tom, že toto dělají pouze některé uzly. Všechny zároveň mohou synchronizační zprávy přijímat. Je to cesta ke zrychlení synchronizace a dosažení úspory energie, ale zároveň zhoršuje přesnost. Může být vhodná pro některé malé, např. domácí sítě, pokud to bude v souladu s jejich požadavky [8].

4.4 RTSP (The Recursive Time Synchronization Protocol)

RTSP algoritmus byl navržen pro vylepšení omezení FTSP protokolu, kterými jsou horší synchronizace vzdálených uzlů a energetická náročnost FTSP protokolu. RTSP protokol zlepšuje přesnost synchronizace a je úspornější díky kombinci metod, jako je časové značení počátku rámce (Start Frame Delimiter time-stamping), minimalizace přenosového zpoždění a nastavování časových značek při každém přeskoku pomocí lineární regerese mezi dvěma body. Jisté optimalizace lze dosáhnout i tím, že uzel, který poskytuje synchronizaci, bude méně často posílat zprávy. Dále navržením modelu tak, že není třeba často resynchronizovat síť, proto je zaveden adaptivní resynchronizační interval. To znamená, že uzel posílá žádost o synchronizaci jen tehdy, pokud je překročen nějaký přijatelný interval odchylky. Zde pokud je skew výrazně větší či menší než 1 a stejně tak, pokud se výrazně liší čas hodin dvou uzlů.

4.4.1 Postup synchronizace

Uzly nejprve vyšlou do broadcastu dotaz na id referenčního uzlu. Místo ve zprávě, které je určené pro id, obsahuje hodnotu -1. Nezáporná hodnota v dalších zprávách poté znamená id nějakého uzlu. Na to uzel čeká určitou dobu, do které většinou obdrží odpověď. Je proveden resynchronizační algoritmus, ten zjistí, zda byla přijata nějaká zpráva, pokud ano, je spuštěn samotný synchronizační algoritmus. Pokud do doby svého timeoutu neobdrží zprávu s id referenčního uzlu a má dostatek energie, vstoupí do výběru nového uzlu tak, že vyšle do broadcastu zprávu s vlastním id.

Typy zpráv

Uzly si mezi sebou vyměňují zprávy typu:

- REQ - žádost o synchroniaci
- ERN - dotaz na id referenčního uzlu nebo pokus o jeho výběr
- REP - odpověď na časovou synchornizaci

Zpracování obdržené zprávy

Pokud obdržel zprávu typu ERN s id jiného uzlu, zachová se podle jeho hodnoty. Pokud je hodnotou -1, znamená to dotaz na referenční uzel. V tom případě pošle odpověď, pokud ví id tohoto uzlu. Zpráva s nezápornou hodnotou v poli id je zpracována jako oznámení nebo pokus o volbu nového referenčního uzlu. Podle velikosti této hodnoty je provedena další akce. Pokud je menší než id referenčního uzlu nebo refereční uzel není známý, pojme odesílatele jako nový referenční uzel a jeho zprávu dále rozpošle. Když je hodnota větší než hodnota id příjemce, vstoupí příjemce do volby o nový referenční uzel rozposláním svého id. Pokud jsou stejné, pouze předá zprávu dál.

V rámci resynchronizačního algoritmu aktuální referenční uzel pravidelně rozesílá tento typ zprávy. RTSP tedy používá broadcast pro udržení povědomí o tom, zda jsou uzly synchronizované a jestli je potřeba vyslat samotnou žádost o synchronizaci - zpráva typu REQ. Uzel má neseřízený čas, pokud přeslechl zprávu typu REP nebo poslal zprávu ERN a zprávu obdržel uzel, který má jiný lokální čas, a derivace frekvenční odchylky vůči odesílateli je větší nebo menší než 1.

Pokud je obdrženou zprávou REQ, poté referenční, nebo už synchronizovaný uzel odpoví se správnou časovou informací. Ostatní uzly si uloží čas odeslání zprávy s časovou informací a lokální čas, kdy byla přijata, rekurzivně pošlou žádost o synchronizaci referečnímu uzlu.

Po přijetí zprávy REP uzel zjistí zpoždění při přenosu, zaznamená hodnotu globálních a lokálních časů. Poté, když obrdží dvě zprávy typu REP, může určit pomocí lineární regrese svou odchylku hodin a frekvenční odchylku.

4.5 TDP (Time-Diffusion-Synchronization Protocol)

V TDP protokolu se uzly sítě dohodnou na určitém referenčním čase a udržují hodiny synchronizované, s určenou maximální odchylkou od tohoto času. V síti se vytváří stromová struktura a existují zde role referenčního kořenového uzlu (master node) a referenčního uzlu druhé úrovně (diffused leader node).

Postup synchronizace je takový, že je rozšířena časová informace směrem od kořenového uzlu k jeho sousedům, z nichž některé se stanou referenčními uzly druhé úrovně. To znamená, že jsou pověřeny šířit časovou informaci dál. Takto postupuje algoritmus ve stromové struktuře. Pokud síť vezmeme jako graf, od kořene k listům.

V návrhu tohoto protokolu je zmíněna ERP procedura [15]. Během ní dochází k opakovanému vybrání obou typů referenčních uzlů. Při volbě bere v úvahu jejich úroveň energie. Jejím účelem je také synchronizovat uzly, které svou odchylkou přesahují maximální stanovenou odchylku v síti. Toho je dosaženo výměnou zpráv s časovou informací.

Zvolený kořenový uzel vyšle svým sousedům broadcastem časovou informaci. Referenční uzly druhé úrovně odpoví kladným potvrzením. Kořenový uzel podle potvrzení, které přijme od uzlu, určí dobu zpoždění při přenosu zprávy tomuto uzlu. Také určí průměrnou dobu přenosu (zpoždění synchronizační zprávy) směrem ke všem svým sousedům a směrodatnou odchylku. Zjištěnou odchylku znovu pošle uzlům v další časově značené zprávě.

Na základě nyní dostupných informací - časové značky, jednocestného zpoždění a směrodatné odchylky už můžou referenční uzly druhé úrovně upravit čas svých hodin a stejným způsobem šířit synchronizaci dál.

4.6 DMTS (Delay Measurement Time Synchronization Protocol)

DMTS protokol je typem protokolu, kde se uzly synchronizují ke zvolenému kořenovému uzlu. Ten vyšle do broadcastu synchronizační zprávu s časem svých hodin a uzly, které ji přijmou, vzhledem k jeho času určí svá zpoždění a synchronizují se.

Na začátku algoritmu je vybrán kořenový uzel, který broadcastuje svůj čas. Zprávu přijmou uzly v jeho dosahu vysílání a nastaví čas svých hodin na čas, který přijaly od kořenového uzlu, a započítají i zpoždění, než se k nim zpráva dostala. Poté samy vyšlou synchronizační zprávu, podle které se synchronizují uzly o jednu úroveň nižší.

DMTS používá systém přidělování úrovní uzlům, přičemž kořenový uzel má úroveň 0, uzly vzdálené jeden přeskok od něj mají úroveň 1 atd. Každý uzel v síti má tedy své číslo, na základě kterého je možné určit jeho vzdálenost od kořenového uzlu. Kořenovým uzlem může být zvolen kterýkoliv uzel v síti, některým z výběrových algoritmů. Nejčastěji to bývá základnová stanice, protože bývá lépe vybavena oproti ostatním senzorům [16].

Přesnost tohoto algoritmu je dána přesností měřicích metod, které uzly používají ke zjištění zpoždění. Tento protokol je flexibilní z hlediska různých topologií a energeticky nenáročný, vzhledem k tomu, že k synchronizaci je nutná pouze jedna zpráva vyslaná každým uzlem. Na druhou stranu není vhodný pro rozhlehlé multi hop sítě. Prioritou tohoto algoritmu je spíše úspora, než přesnost synchronizace [16][9].

5 Zhodnocení vybraných algoritmů z hlediska přesnosti a energetické náročnosti

5.1 Shrnutí RBS

Výhodou RBS je velká přenositelnost a komunikace s mnoha typy hardware i software, které se používají v bezdrátových sítích. Algoritmus nepotřebuje přístup k nízkým vrstvám operačního systému. Dalším plusem RBS algoritmu je odstranění nedeterminismu v čase. Tím, na jakém principu protokol funguje, je zcela eliminován čas, který zabere proces odesílání zprávy, (např. send request MAC vrstvě), což mohou být řádově milisekundy. Tento čas je nedeterministický a závisí na typu procesoru a systémových volání operačního systému. Dále je odstraněna doba čekání k přístupu k přenosovému kanálu, která se liší podle zatíženosti sítě od milisekund až po celé sekundy. Pomocí modifikací operačního systému je možné eliminovat i časovou neurčitost příjmu.

Naopak nevýhodou RBS protokolu jsou nadbytečné zprávy mezi uzly nutné k výměně časových značek. Algoritmus není výhodný pro rozhlehlé sítě s velkým počtem přeskoků. Přesnost algoritmu je dána množstvím času, které trvá uzlu přijmout a zpracovat referenční zprávu [8]. Jistou nevýhodou může být i to, že zařízení, které vysílá synchronizační rámec, není synchronizováno. To lze ale napravit dalšími kroky, kdy roli tohoto zařízení převezme již synchronizovaný uzel.

Použitím RBS lze dosáhnout synchronizace s průměrnou přesností na 29,1 mikrosekund v single hop síti [4].

5.2 Shrnutí TPSN

Protokol TPSN dosahuje lepší výkonnosti než RBS díky časovému značení na MAC vrstvě. Vkládání časových značek probíhá při přenosu a příjmu zprávy, místo posílání zpráv s časovou značkou. Bohužel má jistá omezení v přesnosti, protože nekompenzuje frekvenční odchylku hodin. Tím není robustní proti změnám topologie sítě. Ze stejného důvodu také vyžaduje častou resynchronizaci.

Přesnost se pohybuje okolo 16,9 mikrosekund pro single hop a méně než 20 mikrosekund pro multi hop sítě [4].

5.3 Shrnutí FTSP

Protokol FTSP je robustní vůči výpadkům uzlů a změnám topologie, protože uzly vytvářejí dynamickou topologii a již v rámci samotné synchronizace postupně tvoří strukturu kostry grafu. Je jedním z nejvíce užívaných synchronizačních protokolů ve WSN.

Umožňuje synchronizaci s přesností na 1,48 mikrosekund v single hop sítích a 0,5 mikrosekund v jednom přeskoku v multi hop sítích. Jeho nevýhodou je vyšší spotřeba energie a horší synchronizace vzdálených uzlů [4].

5.4 Shrnutí RTSP

Synchronizací pomocí RTSP protokolu lze dosáhnout přesnosti přibližně 0,3 mikrosekund v jednom přeskoku v rozlehlé multi hop síti. Tím je přesnější než FTSP protokol. Je také úspornější, jeho spotřeba energie je pětinová, uvažujeme-li dlouhou dobu běhu. Algoritmus šetří energii tím, že uzel dočasně ukládá nedávné žádosti o synchronizaci a přeposílá pouze jednu žádost. Na základě odpovědi pak odpoví žádajícím uzlům, což při delší době běhu redukuje počet zasílaných zpráv a šetří zdroj energie. Rovněž metoda adaptivní resynchronizace šetří množství energie [4].

6 Zhodnocení z hlediska bezpečnosti

Popsané protokoly nebyly vyvíjeny s ohledem na bezpečnost před vnějším napadením bezdrátových sítí, nicméně bezpečnost sítě je jednou z jejích požadovaných vlastností, a tak je třeba ji zajistit příslušnými bezpečnostními algoritmy. Následující část se zabývá možnostmi napadení některých z výše popsaných časově synchronizačních protokolů. Útoky na časovou synchronizaci většinou mají za cíl přesvědčit některé z uzlů, že hodiny jejich sousedů mají jinou hodnotu, než je tomu ve skutečnosti. To samozřejmě vede k chybám, jako je synchronizace k nesprávnému času. V protokolech typu sender-receiver útok způsobí více škody, protože se rozšíří po celé síti.

V síti je také možné napadnout energetickou spotřebu, útok spočívá v zasílání falešných signálů a předávání množství dat, která nemusí být vyloženě škodlivá, ale vyčerpávají energii napájecího zdroje tím, že neustále aktivují procesor, vysílač nebo přijímač. Je to jistá forma DoS útoků, to znamená odepření služeb nějakého systému. Zkratka DoS v souvislosti s bezdrátovými senzorickými sítěmi může znamenat i kromě klasického výkladu (Denial of Service) i Denial of Sleep - tedy odepření spánku senzoru.

Bezpečnost ve smyslu odolnosti vůči vnitřním chybám sítě, například výpadku uzlu, je již uvažována při návrhu algoritmu. Z tohoto pohledu je robustní např. FTSP protokol.

6.1 Útoky na časově-synchronizační protokoly

6.1.1 Napadení RBS

V RBS protokolu je broadcastem šířena synchronizační zpráva. Dva uzly či skupiny uzlů si po jejím přijetí synchronizují čas mezi sebou. Útočník může napadnout zprávu s časem např. uzlu A, kterou si vyměňuje s uzlem B, aby narušil jejich synchronizaci. Docílí toho znovu posláním zprávy se starou časovou značkou uzlu A. Zprávu může taktéž zahodit nebo modifikovat.

6.1.2 Napadení TPSN

Útokem je zde obecně napadnutí nekořenových uzlů s nekorektní časovou informací. Čím výše je napadený uzel v hierarchii, tedy blíže kořenovému uzlu, tím horší dopad útoku na synchronizaci, protože chyba se šíří směrem od vyšších do nižších vrstev. V protokolu TPSN lze napadnout několik časů,

1. *T1* - vyslání synchronizačního pulsu
2. *T2* - přijetí synchronizačního pulsu
3. *T3* - vyslání potvrzovacího paketu
4. *T4* - přijetí potvrzovacího paketu

útočník může např. změnit hodnoty časů T2 a T3. Nebo vyrušit inicializační synchronizační puls a znovu ho přehrát po náhodně dlouhém čase. Tento útok způsobí zpoždení v síti, které se přičte synchronizujícím se uzlům, a nelze mu zabránit běžným šifrovacím algoritmem. Dalším možným napadením je modifikace času T4, která také zapříčiní zpoždění.

Uzel může lhát o své úrovni v hiearchii, ostatní uzly budou žádat časovou informaci a dostanou nesprávnou. Také se nemusí zúčastnit fáze přidělování úrovně, což vyřadí na něm závislé uzly.

6.1.3 Napadení FTSP

Základním problémem tohoto algoritmu je, že umožní kterémukoliv uzlu, aby se označil za kořenový uzel po tom, co dlouho neobdržel synchronizační informaci. Napadený uzel se tedy stane kořenovým a ostatní uzly se začnou řídit jeho nekorektní časovou informací, namísto korektní informace od skutečného root uzlu. Výsledkem je samozřejmě chyba synchronizace.

6.2 Obrana proti útokům

Možnou obranou proti výše popsaným útokům na síť je ověřování zpráv a nebo zvýšení počtu synchronizačních zpráv. Cílem je, aby pokud uzel dostane nepravou informaci o čase, byl schopen toto rozpoznat, nepoužil ji a nešířil ji dál. Zajistit bezpečnost obecně stojí nějakou režii navíc, zde zejména ve zvýšení počtu přenášených informací, a tím zvýšení spotřeby energie. Nicméně tam, kde je hrozba napadení sítě, se vyplatí použít některé základní postupy obrany.

Redundance zpráv

Redundance zvyšuje energetickou náročnost provozu, ale má pozitivní vliv na bezpečnost sítě. Použitelná je např. u FTSP protokolu, uzel lépe určí správný čas na základě více časových informací, než pouze od jednoho zprostředkovatele. Takto může být navíc odhalen napadený uzel. Pokud uzel přenáší nesprávná data, zabrání se jejich dalšímu šíření, a tím pádem desynchronizaci. K dalšímu zvýšení bezpečnosti pomůže ověřování zpráv, které je dále popsáno.

Autentikace zpráv

Zvláště u protokolů, u kterých se synchronizační zpráva předává jedním přeskokem, jako je RBS, je třeba ověřit, že odesílatel synchronizační zprávy není napadený uzel. Toho je možné dosáhnout pomocí autentikačního procesu a předáváním soukromého klíče v páru synchronizujících se uzlů.

6.2.1 RTSP

V RTSP protokolu lze předcházet útokům na synchronizaci tak, že každá nová zpráva typu žádost/odpověď na synchronizaci bude doručena po jiné cestě přes různé uzly a v různém čase žádajícímu uzlu. Nově přidaným uzlům se přiřadí identita, pod kterou budou komunikovat v síti, a zpráva se bude považovat za věrohodnou, pouze pokud je od ověřeného uzlu. Na bezpečnosti protokolu přidává také pole zprávy, které obsahuje čas referečního uzlu, takže je možné zamezit šíření příchozí zprávy od uzlu, který by se snažil předat značně rozdílnou informaci.

7 Závěr

Bezdrátové senzorické sítě jsou široce využitelné, distribuované výpočetní systémy, které zároveň mají své specifické požadavky. Ty jsou v mnoha ohledech velmi odlišné od jiných typů bezdrátových či počítačových sítí. V úvodní části práce proběhlo seznámení s jejich vlastnostmi, omezeními, architekturou a komponentami. Implementace těchto systémů často bývá kompromisem mezi úsporou a výkonem. Aby bezdrátová senzorická síť plnila svou funkci, nestačí propojit senzory, ale je třeba vytvořit právě systém, který bude vykonávat koordinovanou činnost.

Spolupráce senzorů vyžaduje např. směrovací algoritmy, ale také časovou synchronizaci. V této práci byla popsána časová synchronizace jednak jako obecné téma, ale především pak v souvislosti se senzorickými sítěmi. Byly představeny různé metody časové synchronizace ve WSN a šest protokolů, které se často ve WSN za tímto účelem používají.

U těchto protokolů byl vysvětlen postup algoritmu a v závislosti na konkrétním typu protokolu další parametry, které je možné hodnotit. Nelze jednoduše říci, který protokol je nejvhodnější, protože vždy záleží na konkrétní síti, její topologii, rozměrech a aplikaci, která ji používá. Obecně ale například platí, že čím menší síť, tím spíše je vhodné použití jednoduchých algoritmů, třeba i s redundancí zpráv.

Časově synchronizační protokoly nejsou vyvíjeny s ohledem na bezpečnost před napadením. Komunikaci v síti je vhodné zabezpečit jinak, např. šifrováním komunikace. V části práce byly rozebrány možné útoky na jednotlivé synchronizační algoritmy a nastíněny metody obrany.

Tato bakalářská práce poskytla úvod do problémů časové synchronizace v bezdrátových senzorických sítích. Proběhlo zhodnocení a srovnání běžně užívaných synchronizačních algoritmů z hlediska přesnosti, bezpečnosti a energetické náročnosti.

Seznam zkratek

WSN (Wireless Sensor Network) bezdrátová senzorická síť

ADC (Analog To Digital Converter) převodník analogového signálu

BLE (Bluetooth Low Energy) technologie bezdrátového přenosu

BS (base station) základnová stanice ve WSN

UTC (Coordinated Universal Time) koordinovaný světový čas

GMT (Greenwich Mean Time) greenwichský střední čas

CRC kontrolní součet

ISM (Industrial, Scientific, Medical) pásmo pro rádiové vysílání

TDMA (Time-division multiple access) časový multiplex

FDMA (Frequency-division multiple access) frekvenční multiplex

CDMA (Code-division multiple access) kódový multiplex

UDP (User Datagram Protocol) protokol pro nespolehlivý přenos

TCP (Transmission Control Protocol) protokol pro spolehlivý přenos

LZW (Lempel-Ziv-Welch) bezeztrátový kompresní algoritmus

LR-WPAN (Low-Rate Wireless Personal Area Network)
technologie bedrátového přenosu

MEMS (Micro-Electro-Mechanical Systems)
mikrosystémy se snímacím zařízením

TCP/IP (Transmission Control Protocol/Internet Protocol)
sada protokolů síťové komunikace

Literatura

[1] *www.ntp.org.*

[2] *https://www.best-microcontroller-projects.com/ppm.html.*

[3] *https://www.researchgate.net/publication/272497022.*

[4] Akhlaq, M. – Sheltami, T. R. *The Recursive Time Synchronization Protocol for Wireless Sensor Networks* [online]. únor 2012. Dostupné z: `https://ieeexplore.ieee.org/document/6166318`.

[5] Akyldiz, I. F. a. M. C. V. *Wireless Senzor Networks.* Wiley, 2010. ISBN 978-0-470-03601-3.

[6] Dargie, W. – Poellabauer, C. *Fundamentals of Wireless Sensor Networks: Theory and Practice* [online]. Wiley, červen 2010. učební text University of Notre Dame, část dostupná z https://www3.nd.edu/ cpoellab/teaching/cse40815/Chapter9.pdf.

[7] Jeff, L. *Clock Synchronization Terminology* [online]. University of New Hampshire, InterOperability Laboratory, červen 2012. Dostupné z: `https://www.iol.unh.edu/sites/default/files/knowledgebase/1588/clock_synchronization_terminology.pdf`.

[8] Lee, H. – W.Yu – Y.Kwon. *Efficient RBS in Sensor Networks* [online]. duben 2006. Dostupné z: `https://ieeexplore.ieee.org/document/1611607`.

[9] Liu, X. – S.Zhou. *Evaluation of several time synchronization protocols in WSN* [online]. 2010. Dostupné z: `https://ieeexplore.ieee.org/abstract/document/5571977`.

[10] M.O.Farooq – Kunz, T. *Operating Systems for Wireless Sensor Networks: A Survey* [online]. květen 2011. Dostupné z: `https://www.mdpi.com/1424-8220/11/6/5900`.

[11] Narayanan, R. P. – tv, S. – Veetil Vineeth, V. Survey on Motes Used in Wireless Sensor Networks: Performance & Parametric Analysis. 2016. zdroj obrázku: https://www.researchgate.net/publication/301523926_Survey_on_Motes_Used_in_Wireless_Sensor_Networks_Performance_Parametric_Analysis/figures.

[12] Čolakov, T. a. D. L. a. L. J. a. i. D. *Bezdrátové senzorické sítě.* EurOpen.CZ, 2009. ISBN 978-80-86583-18-1.

[13] Rucksana, S. – Babu, C. – Saranyabharathi, S. *Efficient timing-sync protocol in wireless sensor network* [online]. březen 2015. Dostupné z: https://ieeexplore.ieee.org/document/7193014/.

[14] Stojmenovic, I. *Handbook Of Sensor Networks: Algorithms And Architectures* [online]. Wiley, 2005.

[15] Su, W. – Akyildiz, I. F. Time-diffusion synchronization protocol for wireless sensor networks. *IEEE/ACM Transactions on Networking*. duben 2005, 13, 2, s. 384–397. ISSN 1063-6692. doi: 10.1109/TNET.2004.842228.

[16] Xiao, Y. *Underwater Acoustic Sensor Networks* [online]. květen 2010. ISBN 9781420067118.

[17] Yener Bülent, F. S. *Time Synchronization in Sensor Networks: A Survey* [online]. [cit.]. Dostupné z: https://pdfs.semanticscholar.org/9814/2e8244bf1830aa38fc21f79e65ce23ff39e4.pdf.