VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

SÍŤOVÁ ARCHITEKTURA A PROPOJOVÁNÍ VESTAVĚNÝCH SYSTÉMŮ

DIZERTAČNÍ PRÁCE
DISERTATION THESIS

AUTOR PRÁCE Mgr. ROMAN TRCHALÍK
AUTHOR

BRNO 2012

Bibliografická identifikace

Jméno a příjmení autora:	**Mgr. Roman Trchalík**
Název disertační práce:	Síťová architektura a propojování vestavěných systémů
Název disertační práce anglicky:	Network Architecture and Interconnecting of Embedded Systems
Studijní program:	Výpočetní technika a informatika
Studijní obor:	Informační technologie
Školitel:	Prof. Ing. MIROSLAV ŠVÉDA, CSc.
Rok obhajoby:	2012
Klíčová slova v češtině:	IEEE 1451, IEEE 802.15.4, vestavěné systémy, WPAN, ZigBee, univerzální aplikační brána, metriky, AODV.
Klíčová slova v angličtině:	IEEE 1451, IEEE 802.15.4, embedded systems, WPAN, ZigBee, univerzal application gateway, metrics, AODV.

Životopis

Osobní informace

Jméno a příjmení:	Roman Trchalík
Datum narození:	6. 9. 1980, Zlín, Česká republika
Email:	trchalik@fit.vutbr.cz

Vzdělání

2005 – nyní	Ph.D. studium, obor Informační technologie VUT v Brně, Fakulta informačních technologií
2003 – 2005	Magisterské studium, obor Aplikovaná informatika Masarykova univerzita v Brně, Fakulta informatiky

Studijní stáže

2008 - 2009	Zahraniční stáž SDU Denmark, Dánsko

Výzkumné projekty

2011 – 2013	Pokročilé bezpečné, spolehlivé a adaptivní IT, VUT v Brně, FIT-S-11-1
2010	Bezpečné, spolehlivé a adaptivní počítačové systémy, VUT v Brně, FIT-S-10-1
2008-2012	ATLANTIS-DeSIRE^2: Dependable Systems International Research and Educational Experience, EC EU, ATLANTIS-DESIRE
2008-2010	Bezpečnost a zabezpečení aplikací sítí vestavěných systémů, GAČR, GA102/08/1429
2007	Průmyslová bezdrátová síť ZigBee, FRVŠ MŠMT, FR2307/2007/G1
2007 – 2013	Výzkum informačních technologií z hlediska bezpečnosti, CEZ MŠMT, MSM0021630528
2007 – 2009	Zvyšování odborné kvalifikace v oblasti bezpečnosti a IP telefonie, CESNET, 215/2007
2006 - 2008	Mezinárodní výukové prostředí pro bezpečnostně kritické řídicí systémy pracující v reálném čase, EC EU, ATLANTIS-ILERT

Akademické zkušenosti

2006 – nyní	Asistent, Fakulta informačních technologií VUT v Brně

Abstrakt

Tato práce se věnuje architektuře vestavěných systémů. Shrnuje současný stav přijatých standardů z rodiny IEEE 1451, které se zabývají vytvářením prostředí pro senzory a jejich zapojení do různých komunikačních sítí. Tyto standardy popisují otevřenou a síťově nezávislou komunikační architekturu pro systém založený na senzorech. Těžištěm práce jsou architektury uvedené jako případové studie, které mohou být využity jako návrhové vzory vestavěných aplikací demonstrované na bezdrátové technologii ZigBee vhodné pro malá zařízení s velmi nízkou spotřebou elektrické energie. Na základě těchto studií jen navržena univerzální brána, která umožňuje aplikační propojení koncových uzlů z různých bezdrátových architektur určených pro senzorové sítě. Práce se dále zabývá modifikováním směrovacího protokolu v síti ZigBee s cílem snížit spotřebu elektrické energie na přenos jednoho datového paketu.

Abstract

The thesis focuses on the architecture of embedded systems. It summarizes the current state of accepted standards from IEEE 1451 family, which deals with creating an environment for the sensors and their involvement in various networks. These standards describe the open, network-independent communication architecture for a sensor-based system. One of the main outcomes of this work is the architectures presented as case studies, which can be used as design patters for embedded applications. They are demonstrated on ZigBee technology suitable mainly for small devices with very low power consumption. Based on these studies the new design of universal gateway was proposed. Its major advantage is that it allows interconnection of the endpoints based on different sensor network technologies. Additionally, the thesis deals with modifying the routing protocol of ZigBee network in order to reduce power consumption required to transmit one data packet.

Klíčová slova

IEEE 1451, IEEE 802.15.4, vestavěné systémy, WPAN, ZigBee, univerzální aplikační brána, metriky.

Keywords

IEEE 1451, IEEE 802.15.4, embedded systems, WPAN, ZigBee, universal application gateway, metrics.

Obsah

1 Úvod

1.1 Motivace

V dnešní době se můžeme setkat s velkým počtem průmyslových nebo domácích aplikací využívajících bezdrátového spojení. Většina těchto aplikací nepotřebuje velkou datovou propustnost sítě, ale místo toho tyto aplikace vyžadují spolehlivou a bezpečnou komunikaci s využitím rádiových systémů, nízkou pořizovací cenu a minimální spotřebu elektrické energie. S požadavky na velkou šířku pásma, která je nutná například pro přenos multimediálních dat, potřebujeme finančně a energeticky náročnější radiové systémy. Avšak převážná většina průmyslových aplikací je nenáročná na služby poskytované danou sítí a vyžaduje pouze základní úroveň adresování pro přenos velmi krátkých zpráv, jako jsou reakce na vzniklé události nebo přenosy informací a dat potřebných pro řídicí činnost.

V principu každé elektronické zařízení, s nímž se v blízké budoucnosti setkáme, bude nejenom inteligentní, ale bude schopné pracovat v globálním digitálním nervovém systému takto propojených inteligentních aplikací. I když takový globální systém je jistě hudbou budoucnosti, je možné již dnes identifikovat odvětví, která usilují o zavádění těchto technologií intenzivněji než jiná. Patří mezi ně obory: bezpečnost, telekomunikace, průmyslová výroba, výroba a dodávky energií, doprava a distribuce, stavebnictví, zdravotnictví, státní správa.

Inteligentní senzory lze zapojit prakticky tam, kde není možné instalovat kabeláž. Jejich velkou výhodou je to, že po uvedení do provozu se samy automaticky spojí do sítí a začlení se do komunikace. Čidla pak mohou monitorovat velmi široké spektrum veličin, jako jsou: tlak, teplota, elektrický proud a napětí, vlhkost, rychlost, pohyb, nebezpečné látky v ovzduší, zámky dveří, detekce plynů, ovládání topení, řízení a kontrola v nepřístupném prostředí apod.

1.2 Cíle práce

Tato disertační práce shrnuje současný stav technologií v oblasti bezdrátových senzorových sítí pro vestavěné systémy a koncepce prostředí pro jejich využití při návrhu řešení systémů. Těžištěm práce jsou architektury uvedené jako případové studie, které mohou být využity jako návrhové vzory vestavěných aplikací. Na základě těchto vzorů je navržena architektura univerzální brány do senzorových sítí, která umožní komunikaci mezi koncovými uzly. Hlavním rysem navrhovaného systému je modulárnost, která umožňuje přidání nového zařízení nebo subsystému bez ovlivnění dalších komponent, resp. stávajícího celku. Zároveň tento návrh řeší zavedení redundance řídicí aplikace a tím zlepšení spolehlivosti celého systému.

V práci je popsána rodina standardů IEEE 1451 a její propojení s různými síťovými architekturami. Z bezdrátových síťových architektur pro vestavěné systémy bude vybrána nejmladší a nejkomplexnější architektura ZigBee a na ní bude otestováno modifikované směrování AODV protokolu, u kterého se omezí všesměrové vysílání paketů kontrolujících lokální konektivitu sousedů. Jako jedna z dalších variant bude představena myšlenka směrování, která je založena na základě spotřebované energie při zaslání zprávy sousedovi.

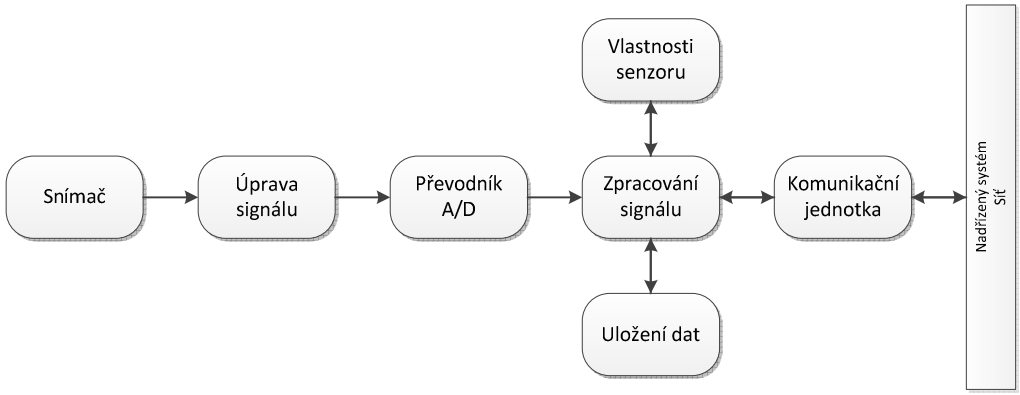
2 Architektura inteligentního rozhraní pro převodníky snímačů a akčních členů

Standardy v oblasti bezdrátových senzorových sítí byly přijaty poměrně nedávno. Proto v této oblasti probíhá intenzivní výzkum a vývoj.

Definice inteligentního snímače [1]:

> *Inteligentní snímač je definován jako snímač, který poskytuje funkce nad rámec nezbytných funkcí pro správné generování zobrazení snímaných hodnot nebo kontrolovaného množství. Tato funkcionalita obvykle zjednodušuje integraci snímače do aplikací v síťovém prostředí.*

Senzor je možné považovat za modulární systém, kde jednotlivé bloky zastávají nějakou činnost, viz Obrázek 1.



Obrázek 1: Model inteligentního senzoru

2.1 Koncept IEEE 1451

Uvedený návrh systému zahrnuje zejména metodu typu *vydavatel/odběratel* pro skupinové (*multicast*) zasílání zpráv s ohledem na bezpečnostní požadavky a nabízí řešení škálovatelnosti a úspornější datový přenos z pohledu moderního Internetu. Předváděné techniky se mohou vzájemně kombinovat s ostatními a mohou podpořit návrh síťových vestavěných aplikací založených na senzorech.

IEEE 1451 [3] se skládá z rodiny standardů, které popisují rozhraní snímačů a akčních členů připojujících se do nadřazené sítě pomocí plug-and-play. Tyto standardy se zabývají návrhem senzoru. Jsou zaměřené na komunikaci mezi převodníkem a mikroprocesorem, na nezávislý objektový model a integraci průmyslových sítí do Internetu. Klíčovým pojmem je definice elektronického katalogového listu TEDS (*Transducer Electronic Data Sheet*), který identifikuje samotný senzor. Obsahuje pojmy jako typ senzoru, jméno výrobce, parametry, kalibrační konstanty a další vlastnosti [8], [22].

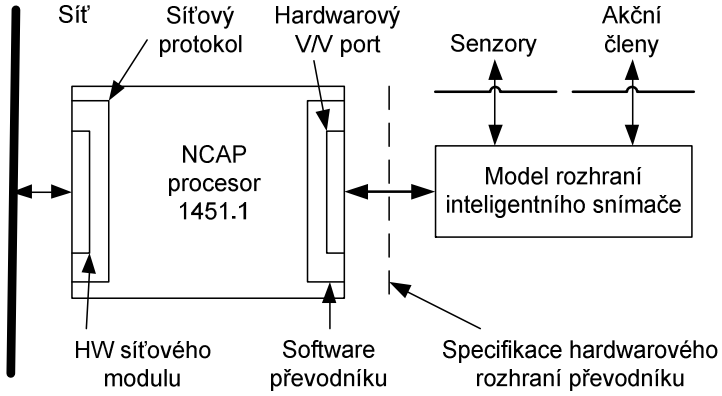
Tyto standardy obsahují zejména: (i) softwarově orientovaný a síťově nezávislý model pro inteligentní senzor, který podporuje aplikační prostředí snímače, (ii) komunikační protokol pro přístup k snímači přes mikroprocesor a (iii) možnost vícebodového připojení senzorů a smíšených komunikačních protokolů pro digitální a analogové snímače.

2.1.1 IEEE P1451.0

Stanovuje základní množiny funkcí, které jsou nezávislé na komunikačních protokolech a médiu. Dále se zabývá koordinováním činností ostatních skupin pro zachování vzájemné kompatibility jednotlivých částí standardu.

2.1.2 IEEE 1451.1

Standard IEEE 1451.1 [6] definuje informační model, který se zabývá objektově orientovanými definicemi síťového aplikačního procesoru (*NCAP*, *Network Capable Application Processor*) a ten je součástí objektově orientovaného inteligentního zařízení připojeného do sítě. Tento model obsahuje definici všech přístupů ke zdrojům a hardwaru snímače na aplikační úrovni. Definice objektového modelu zahrnuje množinu objektových tříd, vlastností, metod a chování. Obsahují stručný popis snímače a okolí síťového zařízení, které pak umožňuje jeho připojení. Více informací je možné nalézt v případové studii [19]. Tento model používá (i) jeden základní blok a základní třídy nabízející vzory pro fyzický blok, (ii) jeden nebo více bloků převodníků, (iii) funkční bloky a (iv) síťové bloky. Základní třídy v sobě zahrnují parametry, akce, události a soubory a poskytují další dílčí třídy, resp. podtřídy.



Obrázek 2: IEEE 1451 – příklad zapojení

Obrázek 2 ukazuje logické zapojení převodníku s NCAP procesorem, který komunikuje se síťovým modulem pomocí síťového protokolu a s blokem převodníku pomocí funkcí v softwarové komponentě.

Standard přináší pohled na senzory zapojené do sítě ve smyslu neutrálního zařízení, ve kterém může být vyvíjena konkrétní implementace [19].

2.1.3 IEEE 1451.2

Tato část standardu popisuje převodník STIM (*Smart Transducer Interface Module*) a komunikační rozhraní (*TII*, *Transducer Independent Interface*) mezi převodníkem a modulem NCAP, které umožňuje připojení snímačů od různých výrobců k NCAP procesoru pomocí elektronických katalogových listů *TEDS* (*Transducer Electronic Data Sheet*). Tato technika je určená pro propojení pouze dvou NCAP procesorů, tzv. *point-to-point* konfiguraci. Původní standard obsahuje komunikaci založenou na sériové SPI lince (*Serial Peripheral Interface Bus*) s hardwarem pro řízení toku dat a časování. Postupně je ale tento standard nahrazen sériovým rozhraním UART (*Universal Asynchronous Receiver/Transmitter*) [37] a USB (*Universal Serial Bus*). Jednobodové spojení lze pak nahradit vícebodovým spojením (*point-to-multipoint*) pomocí sběrnice (viz kap. 2.1.4).

Norma [7] obsahuje osm typů katalogových listů TEDS. Tyto listy jsou do snímače nahrány již při výrobě, což právě umožňuje výše zmíněnou *plug and play* výměnu snímače, přičemž do NCAP procesoru je vždy nahrána správná hodnota. Během modernizace (*upgrade*) snímače tak nedochází k chybám, které by mohly vzniknout při ručním zadávání, zvyšuje se rychlost výměny informací a zlepšuje se správa jednotlivých snímačů.

2.1.4 IEEE 1451.3

Definuje velmi rychlou digitální sběrnici s krátkým dosahem pro spojení senzorů pomocí modulu s rozhraním TBIM (*Transducer Bus Interface Module*). Jedná se vícebodovou verzi předcházejícího standardu IEEE P1451.2. Tzn. k jednomu NCAP procesoru můžeme připojit více snímačů a ke komunikaci s nimi použít protokol HPNA (*Home Phoneline Networking Alliance*) na sdíleném páru vodičů.

Tento standard se nesnaží definovat konkrétní síťový protokol, snaží se pouze specifikovat syntaxi a sémantiku rozhraní mezi objekty a komunikační schéma. Skutečná implementace závisí na použité síti. Existují dva modely síťové komunikace mezi objekty [18]: (i) těsně spojené – komunikace typu klient/server a (ii) volně spojené – komunikace typu vydavatel/odběratel.

2.1.5 IEEE 1451.4

Definuje rozhraní MMI (*Mixed-Module-Interface*) pro smíšenou architekturu senzorů umožňující připojení digitálních i analogových snímačů, které je možné vybavit elektronickými katalogovými listy TEDS. Rozhraní je tvořené dvojvodičovou linkou MicroLan s řízením typu nadřízený/podřízený. Pro komunikaci se může použít dvouvodičové sdílené vedení nebo vícevodičové vedení s vyhrazenými vodiči pro digitální signál.

2.1.6 IEEE P1451.5

Doplňuje do standardu problematiku bezdrátových senzorů, přičemž se zatím jedná o navrhovanou normu [31]. Mezi protokoly, které budou podporované tímto standardem, patří například IEEE 802.11 (WiFi), IEEE 802.15.1 (BlueTooth), IEEE 802.15.4 (ZigBee). Tyto protokoly jsou zavedeny jako typy fyzického rozhraní, viz případové studie [18], [20].

2.1.7 IEEE P1451.6

Návrh standardu pro bezpečnou funkčnost ve vrstvené síťové struktuře s několika řadiči na každé úrovni. Síťová vrstva je tvořena vysokorychlostní datovou sběrnicí typu CAN (*Controller Area Network*), která bývá implementovaná v mikrokontroléru, ale i jako samostatná sběrnice.

3 Bezdrátové technologie WPAN

Personální sítě a sítě s krátkým dosahem WPAN (*Wireless Personal Area Network*) jsou charakteristické větší hustotou pokrytí malého prostoru, malou spotřebou elektrické energie koncovými uzly a větší různorodostí aplikací, které jsou obsaženy v koncových uzlech. Těmito vlastnostmi si senzory a RFID čidla získávají popularitu v různých oblastech komunikace.

Bezdrátová senzorová síť se skládá v prostoru rozmístěných autonomních zařízení obsahující senzory, které spolu navzájem komunikují a monitorují určitý systém. Každé takové zařízení se skládá ze tří částí: (i) snímače, který je schopný změřit nějakou fyzikální, biologickou nebo chemickou veličinu; (ii) jednotky pro zpracování zachycených dat a (iii) systému pro řízení

komunikace. Senzory pracující ve vymezeném prostoru obvykle mívají malou výpočetní sílu a nízkou spotřebu energie, ale celá síť může být poskládána z několika desítek, stovek až tisíců senzorů pokrývající velkou oblast a mezi její hlavní rysy obvykle patří robustnost, spolehlivost a přesnost.

3.1 BlueTooth

BlueTooth patří mezi první technologii, která se dočkala toho, že byla definována standardem IEEE 802.15.1 (r. 1999 verze 1.1) a díky tomu došlo k jejímu rozšíření. Za vývojem BlueTooth stály velké firmy jako Ericsson, IBM, Nokia a Toshiba, což bylo dostatečnou zárukou pro nasazení této technologie na trh. BlueTooth se využívá hlavně v oblasti komunikace mezi zařízeními, jejichž vzdálenost se pohybuje do 10 metrů, přičemž spolu může komunikovat maximálně 8 zařízení [13]. Z praxe pak vyplývá, že při činnosti více než čtyř zařízení dochází k častým kolizím zpráv a výrazně se snižuje celá rychlost sítě. Technologií BlueTooth jsou vybaveny zejména mobilní telefony nebo periferní zařízení počítače jako jsou klávesnice, počítačové myši, mikrofony nebo sluchátka. Komunikace mezi těmito zařízeními probíhá způsobem nadřízený/podřízený, přičemž lze využít i šifrování dat, které využívá 128-bitového klíče a symetrického proudového šifrování. Přenosová rychlost je za ideálních podmínek kolem 1Mbit/s v přenosovém pásmu 2,4 GHz, které obsahuje 79 kanálů o šířce 1 MHz [12].

3.2 Z-Wave

Technologie Z-Wave [27] je stejně jako ZigBee komunikační sada protokolů postavených na otevřené technologii IEEE 802.15.4 využívající stejná vysílací pásma. Obě dvě sítě spadají do kategorie senzorových sítí, kde zařízení jsou obvykle napájena z baterií a neočekává se od nich velký datový tok. Zásadní rozdíl ovšem spočívá ve výrobě komunikačních modulů, kdy Z-Wave rádiové moduly jsou vyráběny pouze jedním výrobcem, kterým je *Sigma Designs*. To samozřejmě zaručuje této technologii bezproblémovou integraci různých zařízení od různých výrobců do jedné sítě.

3.3 ZigBee

Technologie ZigBee [29] je otevřený komunikační standard vyvinutý IEEE organizací (IEEE 802.15.4) a následně podporovaný a rozvíjený ZigBee Alliance (ZBA), jejíž členové jsou přední výrobci elektronických součástek. ZigBee představuje jednoduchou bezdrátovou síť s přenosem dat v pásmu 868 MHz, 915 MHz nebo 2,4 GHz s minimální spotřebou elektrické energie. Přenosová rychlost se pohybuje kolem 20 – 250 kbit/s a vzdálenost mezi dvěma uzly se může pohybovat v řádu několika desítek až stovek metrů. Přenášená data jsou obvykle jednoduché zprávy informující o stavu zařízení, případně o hodnotě naměřené některým ze senzorů. Tato síť může obsahovat tisíce zařízení a díky podpoře topologie typu *mřížka (mesh)* může být signálem pokryta i větší oblast.

4 Adresace uzlů a směrování

Tato kapitola představí modifikaci protokolu AODV pro Ad hoc sítě, která se snaží snížit poměr počtu zpráv směrovacího protokolu k počtu datových paketů přenášených na přenosové lince mezi dvěma sousedy. Dále bude představena myšlenka směrování, která je založena na základě spotřebované energie při zasílání zpráv svému sousedovi [21].

Všechna zařízení v síti ZigBee mají unikátní 64bitovou IEEE adresu, která je po připojení uzlu do sítě koordinátorem redukována na 16bitovou adresu. V síti ZigBee je tedy možné adresovat více než 65 tisíc zařízení (2^{16} uzlů). Těchto redukovaných adres se využívá ke směrování rámců mezi uzly tak, aby se zpráva zaslaná zdrojovým uzlem dostala k cílovému uzlu. Koordinátor je jediný prvek v síti,

který smí přidávat nové uzly do sítě. Avšak může tuto funkci delegovat na jiné *FFD* uzly v síti a zároveň jim přidělit určitý rozsah 16bitových adres. Adresování závisí na síťové topologii. V topologii hvězda se adresa skládá z identifikátoru sítě a identifikátoru cílového zařízení. V jiných topologiích je adresa složena ze zdrojového a cílového identifikátoru.

Problémy v bezdrátovém prostředí:mobilita, omezená šířka pásma, skryté a exponované terminály, spotřeba energie, výdrž baterie.

4.1 Směrování založené na stromové topologii

Senzorová síť ZigBee používá ke směrování svých zpráv k cílovým uzlům algoritmus založený na stromové topologii (*Tree-based routing*), který ohodnocuje cestu k ostatním uzlům. Každý uzel si udržuje směrovací tabulku uzlů a při komunikaci s jiným uzlem si do tabulky vloží nebo obnoví záznam s adresou cílového uzlu a nejbližšího skoku k němu. To znamená uzel, přes který bude zprávu přeposílat. Pokud záznam v tabulce neexistuje, ověřuje se, zda hledaný uzel nepatří mezi některé jeho potomky, což je možné zjistit podle 16bitové adresy hledaného uzlu. V opačném případě je cesta vyhledána pomocí protokolu AODV (*Ad hoc On-Demand Distance Vector*) [30].

4.2 Směrování pomocí AODV protokolu

Protokol AODV (*Ad hoc On-Demand Distance Vector*) je směrovací protokol používaný zejména v bezdrátových Ad hoc sítích, kde uzly nemusí být staticky zasazené v prostoru, ale mohou se různě pohybovat. Jedná se o implementačně jednoduchý protokol s reaktivním přístupem pro získávání cest k cílovým uzlům.

Od roku 2008 bývá tento protokol implementován do zařízení ZigBee. Je výhradně používán u topologií typu mřížka *(mesh)*, případně v topologii hvězda nebo klastrového stromu (*cluster tree*), a to tehdy, pokud dojde ke ztrátě cesty a nelze použít algoritmus směrování založený na stromové struktuře (*tree based routing*).

4.2.1 Princip AODV algoritmu

Princip algoritmu AODV je popsán v dokumentu RFC 3561 [36]. Jedná se o reaktivní protokol, který je rozšířen o proaktivní protokol DSDV (*DSDV, Destination-Sequenced Distance Vector*). Protokol je navržen tak, aby dosahoval lepších výsledků v oblasti vytváření a správy cest, než je tomu u protokolu DSDV [25].

Hlavní cíle protokolu AODV: (i) rozesílat všesměrové pakety jen když potřebujeme najít cestu, (ii) rozlišit správu lokální cesty (konektivity) od celkové správy topologie, (iii) udržovat si informace o dostupných sousedech a (iv) metrika tohoto algoritmu je zaměřena na počet skoků od zdrojového uzlu k cílovému.

AODV snižuje kontrolní režii a minimalizuje množství rozesílaných všesměrových zpráv. V samotném principu AODV vychází z předpokladu, že nalezené linky jsou obousměrné, tzn., že při nalezení cesty z bodu *A* do bodu *B* existuje reverzní cesta z bodu *B* do bodu *A*, která prochází těmi samými uzly, ale v opačném pořadí. Existují však mechanizmy, které dovolují využít princip AODV protokolu i na jednosměrných linkách. Jedním ze způsobů, jak se vyrovnat s jednosměrnými linkami může být například použití černých listin, tzv. *black listů.* Černá listina zakáže použití linky vypuštěním zprávy nebo neúměrným zvýšením některé z metrik.

Každý uzel využívající AODV směrování si udržuje dva čítače: (i) čítač sekvenčních čísel – jednoduchý čítač, který slouží k udržování aktuálnosti reverzní cesty ke zdrojovému uzlu a (ii) čítač RREQ-ID (*Route Request ID*) – čítač, který slouží k identifikaci hledané cesty. K inkrementaci dochází pouze tehdy, když se vytvoří nová žádost o vyhledání cesty tzv. RREQ zpráva (*Route Request Message*).

4.3 Modifikace AODV protokolu

Jednou z výše zmíněných záporných vlastností protokolu AODV je jeho počáteční zpoždění při odesílání paketu. Datový paket musí čekat na to, než se zjistí cesta vedoucí k cíli. Bohužel tuto vlastnost nelze eliminovat. To proto, že patří k vlastnostem reaktivních směrovacích protokolů. Eliminovat lze pouze množství zpráv, které jsou všesměrově rozesílané, jako jsou *RREQ*, *RREP* a *HELLO* zprávy. Hlavním cílem modifikace AODV protokolu je právě snížení velkého množství paketů. Toto snížení můžeme dosáhnout modifikací algoritmu pro získávání cesty tím, že změníme výpočet metriky ovlivňující výběr cesty, po které putují datové pakety. Originální algoritmus používá metriku založenou na počtu skoků mezi zdrojovým a cílovým uzlem. Modifikovaný algoritmus použije pro určení cesty nejrychlejší RREQ zprávu, která dorazí do cílového uzlu. V rámci této změny musí dojít k modifikaci mechanizmu pro údržbu cest, kde se využívá nové speciální zprávy *Alive*. Implementace tohoto modifikovaného algoritmu byla provedena v rámci diplomové práce [5].

4.3.1.1 Proces získávání cest

Proces získávání cesty je stejný jako v originálním algoritmu. Zdrojový uzel zasílá všesměrovou zprávu o nalezení cesty *RREQ* všem ostatním uzlům podle původního algoritmu AODV [36].

Po přijetí *RREQ* zprávy cílovým uzlem dojde k vytvoření odpovědi *RREP* (*Route Replay)*. Oproti původní verzi algoritmu neprochází zprávy *RREP* jinými uzly, které mohou znát cestu a nejsou cílovým uzlem. Cílový uzel odpoví vždy jen na první přijatou zprávu *RREQ*. Předpokládá se, že první přijatá *RREQ* zpráva putovala nejrychlejší cestou s nejmenším počtem skoků, protože na této cestě bude docházet k nejmenšímu zpoždění při zpracovávání zprávy všemi uzly na hledané cestě. Zpoždění vzniklé na přenosovém mediu je ignorováno. Upravený protokol používá metriku prvního přijatého požadavku pro nalezení cesty (*Router Request*).

Zpráva *RREP* se vrací reverzní cestou, po které přišla nejrychlejší *RREQ* zpráva. Jednotlivé uzly na cestě si po průchodu zprávy *RREP* přidávají do svých směrovacích tabulek cestu k cílovému uzlu. Tato cesta je pak použita pro přenos datových paketů.

4.3.1.2 Správa cest

Upravený protokol neobsahuje zprávy informující o chybě linky *RERR* (*Route Error*), ani zprávy *Hello* pro správu lokální konektivity. Místo toho je vytvořena nová zpráva *Alive*, která je periodicky posílána z cílového uzlu zdrojovému uzlu po existující cestě. Tím dochází k udržení živosti aktuálně používané cesty. Po vypršení časovače, tzn. předem definované doby, po kterou uzly neobdrží zprávu *Alive*, dojde k označení cesty za neplatnou. Zpráva *Alive* se posílá jen v situacích, kdy zdrojový uzel posílá data cílovému uzlu a cílový uzel na ně nemusí odpovídat. Z hlediska přenosu dat se v tomto případě jedná o jednosměrnou komunikaci. V případě, že cílový uzel odpovídá datovými pakety zdrojovému uzlu, mluvíme o obousměrné komunikaci a zpráva *Alive* není potřeba na udržení datové cesty.

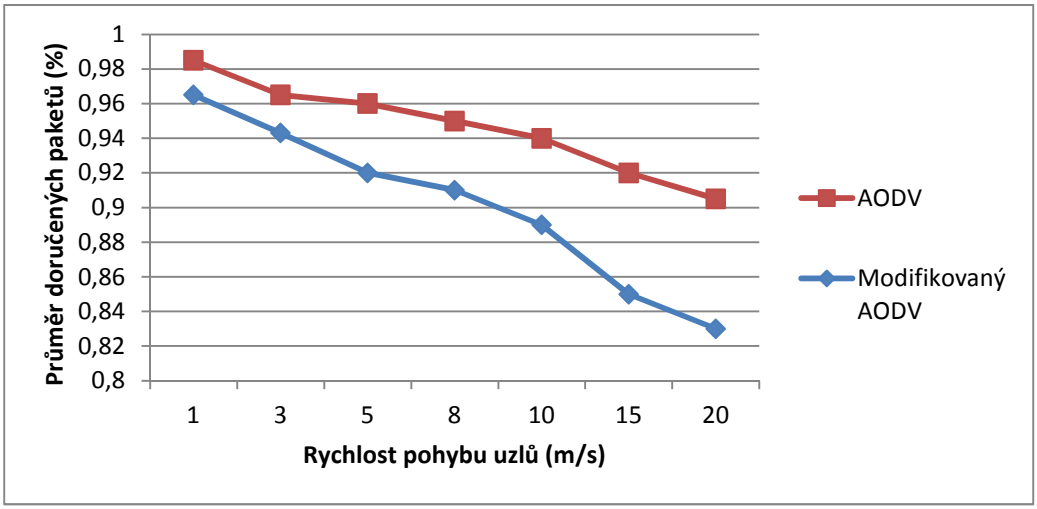
4.3.1.3 Výsledky simulace

Funkčnost modifikovaného protokolu byla simulována v NS-2 (*Network Simulator version 2*) simulátoru. Při zjišťování výkonnostních charakteristik modifikovaného protokolu bylo provedeno několik simulací s různými scénáři. Zkoumány byly vlastnosti jako počet zaslaných kontrolních paketů *Alive*, úspěšnost jejich doručení, normalizovaná zátěž modifikovaného směrovacího protokolu. Výsledky těchto simulací byly porovnávány s protokolem AODV kompatibilním s RFC 3541. Pro každou simulaci byl vytvořen skript, který určuje jednotlivé parametry simulace: rozmístění a počet uzlů v prostoru včetně jejich pohybu a definice uzlů jako zdroje dat nebo cílových uzlů pro příjem dat. Původní implementace [5] byla v rámci této disertační práce přenesena do poslední verze NS-2 2.35 a zároveň byla rozšířena sada simulačních testů.

Úspěšnost doručení paketů

Cílem této simulace bylo zjistit, jak jsou pakety doručovány v podmínkách mobilního prostředí. Simulace byly provedeny při rychlostech uzlů 1, 3, 5, 10, 15 a 20 m/s. Každá simulace byla provedena 10krát s dobou trvání 500 sekund v různých topologiích a náhodně definovaným pohybem uzlů. Zdroje dat byly pro všechny simulace stejné. Výsledné hodnoty simulací byly zprůměrovány, viz Obrázek 3.

Simulace byla provedena na ploše 750x750 m s 50 uzly. Vysílací dosah uzlu byl nastaven na 240 metrů. Zdrojem dat bylo 10 uzlů, které zasílaly 4 zprávy za 1 sekundu.

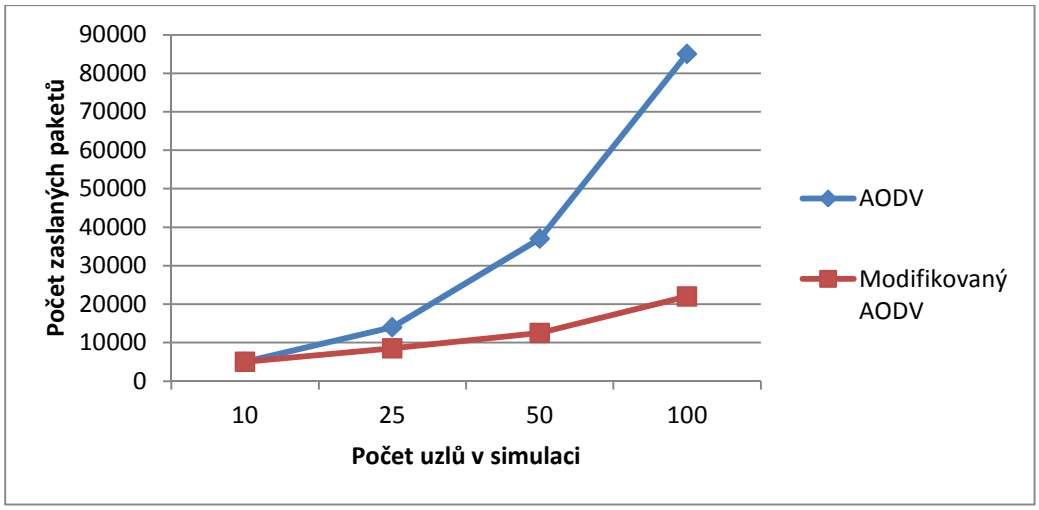


Obrázek 3: Úspěšnost doručení paketů

Z grafu je patrné, že s rostoucí rychlostí pohybu uzlů klesá procento doručených paketů oproti původnímu AODV protokolu. Důvodem tohoto poklesu je častá změna topologie, jejímž důsledkem dochází ke ztrátě linky a poté ke ztrátě paketu během přenosu. AODV protokol vykazuje nižší procento ztráty paketů díky zprávám informujících o chybě linky. Ztráta paketů u klasických TCP/IP sítí se řeší na transportní úrovni v případě TCP protokolu nebo aplikační úrovni v případě UDP protokolu. Senzorová síť typu ZigBee transportní vrstvu neobsahuje, proto tuto ztrátu paketu musí řešit samotná aplikace např. sledováním potvrzovacích zpráv.

Počet zpráv směrovacího protokolu

Druhou sledovanou metrikou je počet poslaných zpráv modifikovaného protokolu. Do výpočtu se zahrnul každý přenos zprávy mezi dvěma sousedními uzly



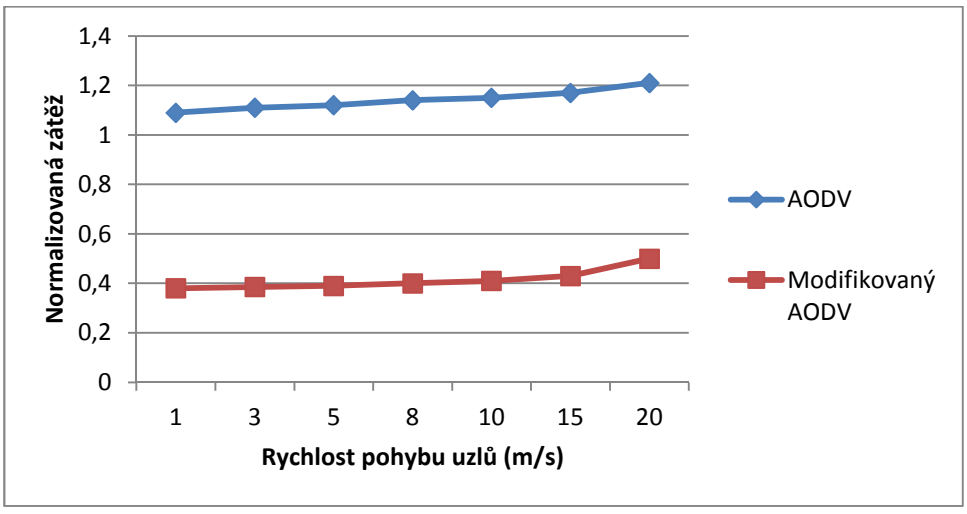
Obrázek 4: Počet zaslaných zpráv

Z principu AODV protokolu je zřejmé, že počet zpráv s rostoucím počtem uzlů roste. Modifikovaný protokol z hlediska počtu zaslaných zpráv dosahuje lepších výsledků díky eliminaci zpráv pro zjišťování lokální konektivity a zpráv oznamujících chybu linky. Menší počet zpráv se tedy projeví v menším zatížení sítě a také menší spotřebou energie jednotlivých uzlů.

Simulace byly provedeny na plochách 350x350 m s 10 uzly, 500x500 m s 25 uzly, 750x750 m s 50 uzly a 1000x1000 m se 100 uzly. Rychlost pohybu uzlů je do 5 m/s. Při simulaci bylo použito 10 zdrojových uzlů. Ostatní parametry zůstaly stejné.

Normalizovaná zátěž

Normalizovanou zátěží se rozumí poměr počtu zpráv směrovacího protokolu a počtu datových paketů přenášených na přenosové lince mezi dvěma sousedy. Jinými slovy se jedná o velikost režie, kterou musí síť vynaložit, aby byl doručen jeden datový paket.



Obrázek 5: Normalizovaná zátěž

S rostoucí rychlostí uzlů stoupá průměrné množství zpráv směrovacího protokolu, které je nutné pro doručení datového paketu. U modifikovaného protokolu je toto množství výrazně nižší než u originálního AODV protokolu. Tato skutečnost je významně ovlivněna eliminací kontroly linek.

Shrnutí

Modifikovaný protokol AODV oproti původnímu AODV protokolu významným způsobem redukuje množství zasílaných zpráv mezi jednotlivými uzly a významně tak snižuje normalizovanou zátěž na jednu zasílanou zprávu. Na druhou stranu negativně snižuje množství doručených paketů, což zvyšuje nároky na aplikační část v uzlu, která se se ztrátou paketu musí vypořádat.

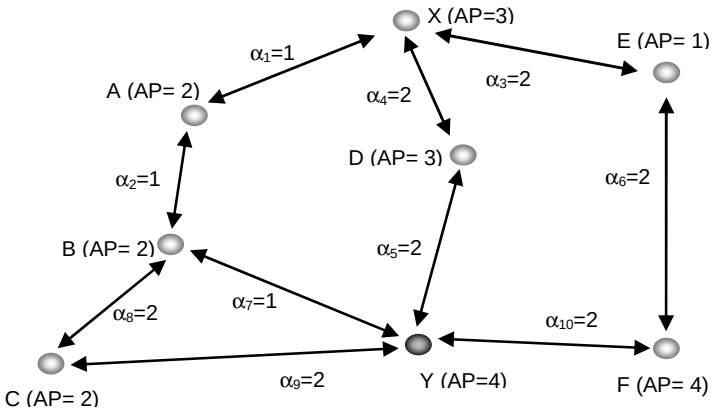
4.4 Energeticky efektivní směrování

Tradiční techniky směrování typu Ad hoc nejsou obvykle vhodné pro požadavky senzorových sítí. Síťové vrstvy senzorových síti jsou obvykle navrhovány podle zásad, které říkají, že [21]:

- Důležitým faktorem je vždy energetická účinnost.
- Data by se měla převážně uchovávat v hlavním uzlu (*data-centric*).
- Měla by se provádět agregace dat, pokud to neovlivňuje spolupráci jednotlivých uzlů.
- Ideální senzorová síť má mít informaci o umístění svých uzlů a energetické náročnosti jednotlivých cest, na základě kterých je možné určit vhodnou cestu.

Energeticky efektivní směrování může být založeno na dostupné energii *AP* (*Available Power*) na každém uzlu a energii α, která je nutná pro přenos dat mezi dvěma sousedními uzly. Pro zaslání zprávy z uzlu *X* do uzlu *Y* (viz Obrázek 6) získáme několik možných cest, po kterých může zpráva putovat:

- *Cesta 1*: *X-A-B-Y*, celková *AP* = 4, celková α = 3
- *Cesta 2*: *X-A-B-C-Y*, celková *AP* = 6, celková α = 6
- *Cesta 3*: *X-D-Y*, celková *AP* = 3, celková α = 4
- *Cesta 4*: *X-E-F-Y*, celková *AP* = 5, celková α = 6



Obrázek 6: Směrování na základě energické náročnosti

Směrování je pak zvoleno na základě jedné ze strategií:

- Minimální dostupná energie na cestě – vybere cestu, na které spotřebuje co nejméně energie na přenos paketu mezi zdrojovým uzlem *X* a cílovým uzlem *Y*. Tzn. celková suma spotřebované energie α je minimální. To odpovídá cestě 1.
- Maximální dostupná energie na cestě – vybere cestu, která má maximální celkovou *AP*. Celková *AP* je dána součtem *AP* energií všech uzlu na zvažované cestě. To odpovídá cestě 2.

- Minimální počet skoků na cestě – je preferovaná cesta s minimálním počtem uzlů. Algoritmus vybere stejnou cestu jako algoritmus minimální dostupné energie, jestliže je na cestách stejná spotřeba energie α. To odpovídá cestě 3.
- Maximum z minimálních dostupných energií – je taková cesta, kde minimální dostupná energie *AP* u každého uzlu na cestě je větší než minimální dostupná energie *AP* u ostatních uzlů na jiných cestách. Cesta 3 je efektivnější než cesta 1. Tento algoritmus vylučuje riziko použití uzlu s nízkou hodnotou dostupné energie neúměrně dříve, než u ostatních uzlů. Použije se cesta, jejíž uzly mají mnohem větší dostupnou energii *AP*.

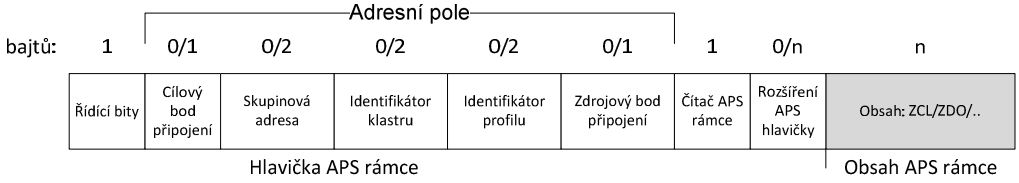
5 Případové studie

Cílem této kapitoly je představit dvě řešení pro připojení bezdrátové senzorové sítě ZigBee k Internetu, které mohou být použity jako návrhové vzory. Kvůli odlišnostem architektur senzorových sítí a Ethernetu jsou předváděná řešení založená na aplikačních branách. První řešení je založené na aplikační bráně s vlastním jednoduchým komunikačním protokolem ze strany Internetu. Druhé řešení je založeno na standardním protokolu SNMP. Uvedená řešení byla experimentálně implementována v rámci diplomových prací [4] a [14].

5.1 Zasílání zpráv a párování

Pro *přímé adresování* aplikačního objektu musí zdrojová aplikace znát adresu uzlu, číslo bodu připojení (*Endpoint*) a číslo klastru (*Cluster*). Z tohoto důvodu ZigBee síť podporuje dva způsoby adresování aplikací a služeb. První způsob je složitější a nazývá se *přímé adresování* (*Direct Addressing*). Pro přímé adresování musí zdrojový uzel znát adresu cílové aplikace. Pokud ji nezná, musí si ji zařízení zjistit. Nevýhodou tohoto způsobu adresování je to, že k posílání přímých zpráv je potřeba velká režie obsažená v objevování okolních zařízení a jejich aplikací, správě a uchovávání těchto adres v paměti. Tento typ adresování zvyšuje energetickou spotřebu uzlů.

Druhým a zároveň lehčím způsobem je *nepřímé adresování* tzv. *párování* (*Binding*), při kterém se vytváří logická vazba mezi jednotlivými aplikačními objekty. Koordinátor sítě si pak udržuje tabulku záznamů, ve které si páruje klastry nebo body připojení. Při vytvoření páru mohou daná zařízení spolu komunikovat přes koordinátora, který na základě identifikace zdrojového uzlu, ID klastru určí příjemce nebo v případě použití bodů připojení (*Endpoint*) určí skupinu uzlů (*Group Addressing*) [30]. Na následujícím obrázku se nachází formát aplikačního rámce s vyznačenými políčky, která slouží k adresaci aplikace.



Obrázek 7: Formát aplikačního rámce (APS)

5.1.1 Aplikační profil

Aplikační profil je logická vlastnost softwarové komponenty tzv. aplikačního objektu a jeho rozhraní. Ve své podstatě je tento profil založen pouze na dohodě mezi výrobci, kteří své aplikace opatřují smluveným 16bitovým identifikátorem profilu (*Profile ID*), aby bylo možné poznat, o jaký druh komponenty se jedná. Hlavním důvodem pro zavedení tohoto profilu je umožnění vzájemné

komunikace mezi dvěma zařízeními různých výrobců. Např. pod identifikátorem pomocí profilu *Automatizace domácností* (*Profile ID = 0x0104*) si může koncový zákazník koupit bezdrátový vypínač a světlo od různých výrobců, přičemž má zajištěno, že budou spolu komunikovat a splní svůj účel. Každý profil obsahuje klastr, který specifikuje komunikační rozhraní.

Rozlišujeme tři typy profilů: (i) veřejný, který je spravován ZigBee Aliancí; (ii) privátní profily, které mohou být definovány výrobcem a (iii) publikované profily, které vznikají zveřejněním privátních profilů [2].

5.1.2 Klastr

Klastr (*Cluster*) je skupina atributů a příkazů, které společně definují komunikační rozhraní mezi dvěma aplikacemi. Toto komunikační rozhraní je založeno na architektuře klient/server a má svůj unikátní identifikátor *Cluster ID*. Klastr slouží jako vstupní nebo výstupní rozhraní pro konkrétní aplikační objekt, resp. pro jeho bod připojení (*Endpoint*).

Pokud navážeme na předchozí příklad se zakoupeným bezdrátovým vypínačem a světlem, tak vstupním klastrem pro světlo je pouze atribut *OnOff*, který může být prezentován hodnotou 0 (vypnout) a 1 (zapnout). Vstupním klastrem je proto, že hodnotu pouze přijímá a žádnou odpověď nevrací nazpět. Pro vypínač to bude přesně opačně, tzn. atribut *OnOff* bude výstupním klastrem bodu připojení.

5.1.3 Bod připojení

Bod připojení (*Endpoint*) je seznam klastrů, které jsou podporované aplikačním objektem v rámci aplikačního profilu. Pro každý takový klastr musí být definován jednoduchý deskriptor (*Simple Descriptor*), který klastr identifikuje a zároveň je součástí každého aktivního bodu připojení (*Endpoint*) na zařízení.

Jednoduchý deskriptor pro své zařízení obsahuje seznamy: (i) vstupních klastrů s podporovanými výstupní klastry typu server a (ii) výstupních klastrů s podporovanými klastry typu klient.

5.1.4 Knihovna systémových klastrů

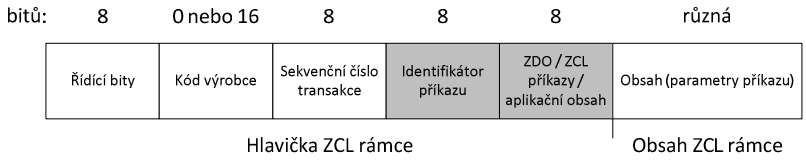
Knihovna systémových klastrů obsahuje popisovače zařízení (*Device Descriptors*) a funkce ze skupiny ZigBee objektů ZDO (*ZigBee Device Object*). Tyto popisovače a funkce poskytují základní informace o daném zařízení. Existuje několik typů těchto funkcí a popisovačů a povinnost jejich implementace se odvíjí od role daného uzlu (koordinátor vs. směrovač vs. koncový uzel). Mezi nejzákladnější patří:

- Popisovač uzlu (*Node Descriptor*) – popisuje logický typ zařízení (koordinátor, směrovač, koncový uzel), frekvenční pásmo, apod.
- Popisovač napájení (*Power Descriptor*) – sděluje informace o typu napájení (permanentní vs. baterie), v případě baterií i jejich stav.
- Jednoduchý popisovač (*Simple Descriptor*) – pro každý bod připojení (*endpoint*) obsahuje identifikátor aplikačního profilu a podporované vstupní a výstupní klastry.
- Popisovač aktivních bodů připojení (*Active Endpoints*) – získává seznam všech aktivních bodů připojení, které dané zařízení obsahuje.
- Správce okolních zařízení (*Management Network Discovery*) – používá se na skenování sítě, zařízení sdělují informace o identifikátoru sítě, vysílacích kanálech, verzi ZigBee protokolu apod.

Formát datového rámce popisovačů a funkcí je určen specifikací ZigBee [30] a je zasazen do APS rámce, viz Obrázek 7.

5.1.5 Knihovna ZigBee klastrů

Knihovna ZigBee klastrů (*ZCL, ZigBee Cluster Library*) je uložiště pro aplikační klastry definované a spravované ZigBee aliancí. Výrobci nových zařízení si vybírají vhodná čísla klastrů ze ZCL, kde jsou rozděleny do generických a aplikačně orientovaných doménových skupin [28]. Pro vyvolání operace na senzoru se používá generický ZCL rámec (Obrázek 8), jehož obsah je dekomponován a příkazy s parametry jsou předány aplikaci.



Obrázek 8: Generický formát ZCL rámce

5.2 Aplikační brána - interpret na koordinátoru

Návrh této aplikační brány je vytvořen na základě dostupného zařízení, vývojové desky Picdem Z od firmy Microchip. Jedná se o vývojové sady typu plně funkčního zařízení (*FFD*) s teplotním senzorem, dvěma tlačítky a dvěma diodami. Kromě bezdrátového rozhraní ZigBee tato zařízení obsahují sériové rozhraní typu RS-232 pro propojení s počítačem nebo jiným zařízením. Vzhledem k rozdílnosti komunikačních protokolů TCP/IP a protokolu ZigBee je potřeba informace mezi těmito sítěmi překládat. Návrh aplikační brány musí vzít v úvahu velikost paměti na vývojových deskách, kdy do 64kB se musí vejít přeložené zdrojové kódy ZigBee vrstev, ale i aplikace, která bude plnit funkci požadované aplikační brány.

5.2.1 Architektura

Při návrhu této brány hlavními požadavky jsou:

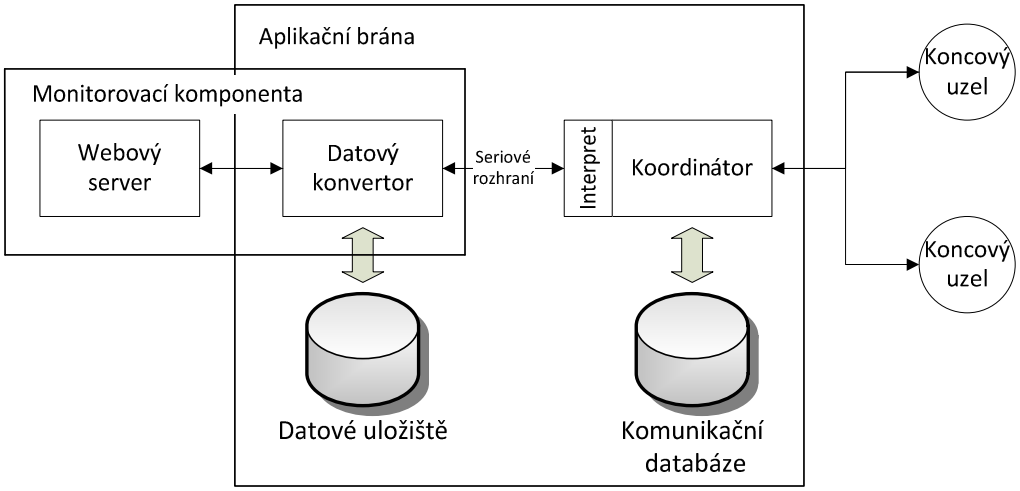
- Zachovat originální funkčnost senzorové sítě.
- Monitorovat stav sítě z Internetu nebo Intranetu.
  - Identifikovat jednotlivé uzly sítě – rozpoznat jejich adresy.
  - Identifikovat na uzlech jejich senzory a akční členy.
  - Získávat hodnoty ze senzorů.
- Umožnit měnit stav nebo hodnoty na koncových uzlech z Internetu/Intranetu.
- Zavést univerzální rozhraní.
- Možnost využití tohoto rozhraní u jiných technologií.
- Jednoduchost.

Návrh této aplikační brány, vzhledem k nedostatkům vývojových sad pro ZigBee, je rozdělen na dvě části, viz Obrázek 9.

- Koordinátor, resp. jeho aplikační část, která zajišťuje:
  - Originální funkčnost sítě – provádí sestavení sítě, připojování a odpojování uzlů, šifrování zpráv, ověřování integrity zpráv apod.

- Interpretaci přijatých zpráv a stavů sítě do jednoduchého proprietárního jazyka, kterému rozumí monitorovací komponenta.
- Zajišťuje uchovávání síťové topologie.

- Monitorovací komponenta s datovým konvertorem, která zajišťuje:
  - Datovou konverzi – převody čísel.
  - Ukládání a přípravu dat pro webový server.
  - Vyzvednutí požadavku ze strany webového serveru a jeho přeposlání na koordinátor.



Obrázek 9: Architektura komunikační brány – interpret na koordinátoru

5.2.1.1 Monitorovací komponenta

Monitorovací komponenta je složená ze dvou částí, a to z datového konvertoru a řídícího serveru, např. webového. Datový konvertor se stará o výměnu dat mezi upravenou aplikací koordinátoru a webovým serverem. Zajišťuje řádné předávání hodnot a příkazů a také jejich správné uložení do datového rámce proprietárního protokolu. Webový server zobrazuje uživateli získané hodnoty senzorů ze sítě ZigBee. Pokud mají aplikace těchto senzorů definován vstupní klastr obsahující atributy pro změnu nastavení, je možné jejich hodnoty měnit přes webové rozhraní.

5.2.1.2 Registrace do sítě

Při procesu registrace uzlu do sítě si koordinátor vyžádá od nově registrovaného uzlu seznam všech jeho aktivních bodů připojení (*Endpoints*), a to pomocí dotazu na jeho jednoduchý deskriptor (S*imple Descriptor*), který musí být implementován na všech zařízeních. Přidávaný uzel zasílá koordinátoru seznam svých bodů připojení včetně jejich vstupních a výstupních klastrů. Získané informace o bodech připojení jsou uloženy v datové struktuře koordinátora.

Při registraci a sestavování sítě provádí koordinátor alokaci frekvenčního pásma, generování identifikátoru sítě *PanID* a poté pomocí aplikační podvrstvy vrstvy a služeb ze ZDO se připojuje jako první prvek sítě. Po přijetí potvrzení o úspěšném sestavení sítě z APS vrstvy dochází k vytvoření základního kořene datové struktury s informacemi o identifikátoru sítě *PanID*, vysílacím kanálu, době jeho skenování a zároveň se vytváří seznam pro uzly v nové síti. Po sestavení sítě se automaticky spouští proces přidání koordinátora do sítě a tím se zároveň vloží první uzel do seznamu uzlů *NodeInfo* s indexem 0. Poté se zjišťují základní informace o adrese uzlu a typu zařízení (koordinátor, směrovač, koncový uzel) apod. Pak se zjišťují informace o existujících bodech připojení na uzlu. Informace se ukládají do datové struktury na koordinátora.

5.2.2 Shrnutí

Navržené řešení umožňuje monitorování stavu sítě a jednotlivých uzlů s jejich senzory. Hlavním kladem tohoto systému je jeho jednoduchost a implementační volnost, která dovoluje programátorovi vytvořit i nestandardní řešení. Tato volnost je však vykoupena rozšiřováním funkčnosti senzorové sítě v tom smyslu, že při přidání uzlu s novým aplikačním profilem (např. topení) se musí upravit aplikační část na straně koordinátora, datového konvertoru i webového serveru. Zejména úprava softwaru na koordinátoru vyžaduje jeho vypojení ze sítě kvůli nahrání nového firmwaru, což způsobí omezení funkčnosti vestavěného systému po dobu aktualizace jednotlivých prvků systému.

5.3 Aplikační brána – SNMP

Protokol SNMP (*Simple Network Management Protocol*) patří mezi nejpoužívanější protokoly pro správu zařízení připojených do sítí typu TCP/IP. Patří mezi protokoly aplikační vrstvy a je obohacený datovým modelem a definicemi datových objektů.

Jádro protokolu tvoří skupiny příkazů typu *get* a *set*, které umožňují klientské aplikaci (správci, manažerovi) zjistit nebo změnit hodnotu atributu. Např. počet otáček ventilátoru, aktuální využití procesoru nebo paměti, stav baterií apod. Protokol SNMP zároveň umožňuje přenášet asynchronní zprávy od sledovaného zařízení k monitorujícímu zařízení, jedná se o tzv. TRAP zprávy.

Každá sledovaná hodnota na sledovaném zařízení se v rámci SNMP komunikace jednoznačně identifikuje pomocí číselného identifikátoru OID (*Object Identifier*). Identifikátor OID je tvořen posloupností čísel oddělených tečkou, která představuje konkrétní úroveň stromové struktury, do které jsou OID mapovány. Každá společnost a každé její zařízení podporující SNMP protokol má své unikátní mezinárodně přidělené číslo.

Sledované atributy jsou definovány a spravovány v databázích MIB (*Management Information Base*). MIB databáze jsou podobné standardním databázím v tom smyslu, že popisují strukturu a zároveň i formát dat. MIB moduly jsou definované podle pravidel SMI (*Structure of Management Information*), které jsou popsány v dokumentech RFC1155 [32], RFC1212 [33] a RFC1213 [34] a RFC1215 [35].

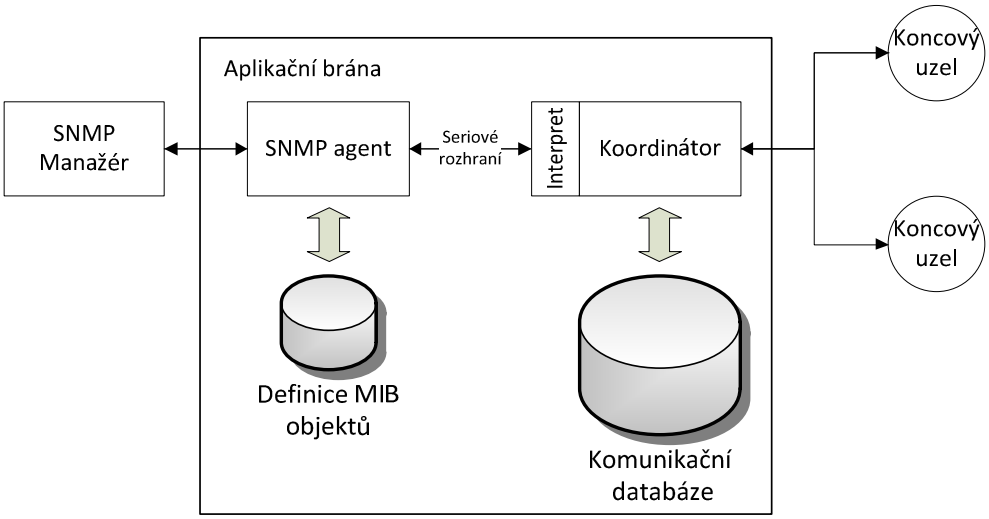
MIB databáze je hierarchická stromová struktura, která odpovídá danému konkrétnímu zařízení s objektově orientovaným modelem jako sada SNMP objektů, relací a operací mezi objekty. Zároveň slouží k překladu číselných řetězců OID do textové podoby.

5.3.1 Architektura

V klasických TCP/IP sítích se standardně umisťuje agent přímo na sledované zařízení, tzn. na směrovač, přepínač nebo klientskou stanici. Správa těchto zařízení probíhá centrálně z aplikace SNMP manažera. Výhoda spočívá v jednoduché správě těchto zařízení, pokud dojde ke změnám struktury sítě. Jednotlivá zařízení se mohou jednoduše odpojovat nebo nová zařízení přidávat. Prakticky většina nastavení se provádí na SNMP manažerovi, protože množina spravovaných objektů se u stejných typů zařízení nemění. Každému objektu OID z databáze mohou být napevno přiřazeny funkce pro získávání a změny hodnot.

Vzhledem k tomu, že implementace agenta protokolu SNMP je poměrně jednoduchá, nabízí se možnost přeložit agenta přímo pro jednotlivé koncové uzly ZigBee. SNMP manažer komunikuje pouze v IP sítích, proto by se musel upravit koordinátor sítě ZigBee tak, aby prováděl mapování adres uzlů ZigBee a IP adres. Tento přístup je však příliš komplikovaný. Pro každý koncový uzel by se

musel kompilovat speciální software, čímž se stává tato myšlenka v praxi nereálná, protože ZigBee síť může obsahovat i několik tisíc uzlů. Další negativní vlastností tohoto řešení je ztráta možnosti použít zařízení různých výrobců.



Obrázek 10: Architektura SNMP systému pro správu ZigBee sítě

Druhou variantou použití protokolu SNMP je implementovat agenta přímo do koordinátoru sítě ZigBee nebo na počítač, ke kterému bude koordinátor připojen. Protože nebyl k dispozici ZigBee koordinátor s Ethernetovým rozhraním, agent SNMP byl umístěn na počítač, který byl propojen sériovou linkou s koordinátorem. Při vhodném navržení MIB objektu s tabulkami je možné vyřešit problém dynamické změny záznamů, které vznikají připojováním a odpojováním uzlů ZigBee sítě. Architektura tohoto návrhového vzoru je zobrazena na následujícím obrázku 10.

Úprava na straně koordinátora vyžaduje implementaci modulu pro zpracování asynchronních zpráv přicházejících ze strany sériového portu. Tzn. zpracování příkazů ze strany SNMP manažera, např. zjišťování hodnot jednotlivých atributů nebo jejich změna. Koordinátor musí zároveň vyhodnotit, jestli je zpráva určená jemu nebo je nutné o hodnotu atributu požádat některý koncový uzel nebo skupina uzlů.

5.3.2 Shrnutí

Navržené řešení nabízí oproti předchozímu návrhu (kap. 5.2) univerzální rozhraní pro komunikaci se senzorovou sítí pomocí protokolu SNMP. Díky tomu je možné pro ovládání sítě použít libovolného SNMP manažera. Pomocí skriptů je pak možné přesunout řídící logiku nebo aspoň její část na vzdálený počítač. Použití SNMP agenta na straně koordinátoru zároveň umožňuje zavedení redundantního SNMP manažera v případě aktualizace nebo výpadku primárního. Nevýhodou je opět aktualizace firmware na straně koordinátora při přidávání nového aplikačního profilu.

6 Univerzální aplikační brána

V oblasti administrace s rostoucím počtem zařízení se firmy snaží o zjednodušení správy zařízení a zefektivnění práce administrátorů těchto zařízení. U systémů s různými síťovými technologiemi rostou výrazně nároky na jejich správu a jejich případnou integraci do informačního systému.

Pro průmyslové bezdrátové sítě v dnešní době existují tři významnější bezdrátové technologie, které dovolují vytváření pokročilejších síťových topologií. Jsou to např. ZigBee, Z-Wave nebo BlueTooth.

6.1 Architektura

Brány (*Gateways*) a linkové mosty (*Bridges*) nabízejí dvě různé techniky, jak propojit různé heterogenní sítě. Brány poskytují plnou konektivitu se všemi vlastnostmi a jsou implementovány na různých zařízeních. Linkové mosty jsou jednodušší a vyžadují mnohem menší místo pro aplikaci. Linkové mosty jen převádějí datový rámec z jednoho média na druhé. Naopak brány převádějí bezdrátové protokoly a data ze senzorových sítí do různých formátů, které jsou nezbytné pro komerční, průmyslové a rezidenční systémy. Příkladem může být *BACnet* a *LonWorks* pro rezidenční systém, *SCADA* a *Modbus* pro průmyslové sítě a HTML s XML jazykem pro aplikace na Internetu [3], [8], [9] a [10].

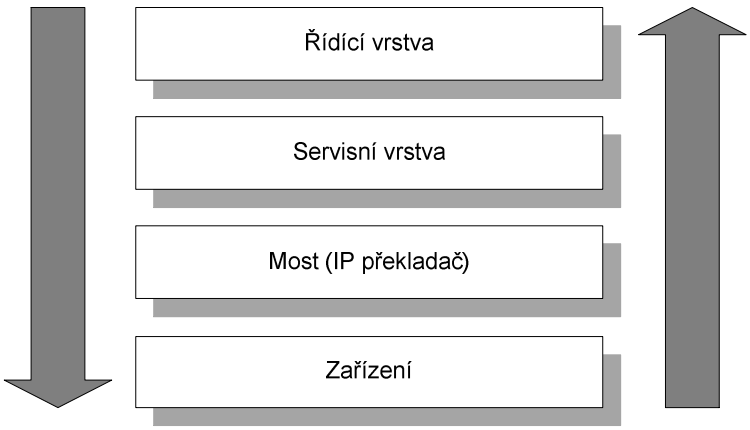
6.1.1 Požadavky na univerzální bránu

Požadavky na univerzální bránu byly definovány následovně:

- Nezávislost na architektuře senzorové sítě – povýšení (abstrakce) řídícího modulu nad jednotlivé síťové architektury zvětšuje prostor z hlediska rozšíření funkčnosti systému.
- Jednoduchá instalace – zjednodušení instalace jednotlivých zařízení pro různé senzorové sítě, šetří se tím náklady a snižuje se tím pravděpodobnost vytvoření chyby při instalaci.
- Modulárnost – systém musí umožnovat přidání nového zařízení nebo subsystému bez ovlivnění dalších komponent (zařízení), která jsou již součástí systému. Zároveň aplikační část systému musí být schopna přijmout nové zařízení a začít s ním bez větších úprav pracovat.
- Škálovatelnost – systém by měl umožnovat správu několika set zařízení, tak aby vyšší počet zařízení neovlivňoval výkon jednotlivých zařízení.
- Bezpečnost – systém by měl obsahovat základní zabezpečovací mechanismy autentizace proti zneužití systému běžnými uživateli a proti síťovým útokům.

6.1.2 Vrstvený model

K návrhu univerzální aplikační brány do senzorových sítí je potřeba zavést virtuální komunikační vrstvu založenou na principu vydavatel/odběratel, kdy aplikační logika systému se bude registrovat u senzorové sítě k odebírání zpráv. Z hlediska funkčních požadavků lze základní návrh univerzální brány rozdělit do 4 vrstev [24], viz Obrázek 11.



Obrázek 11: Vrstvený model brány

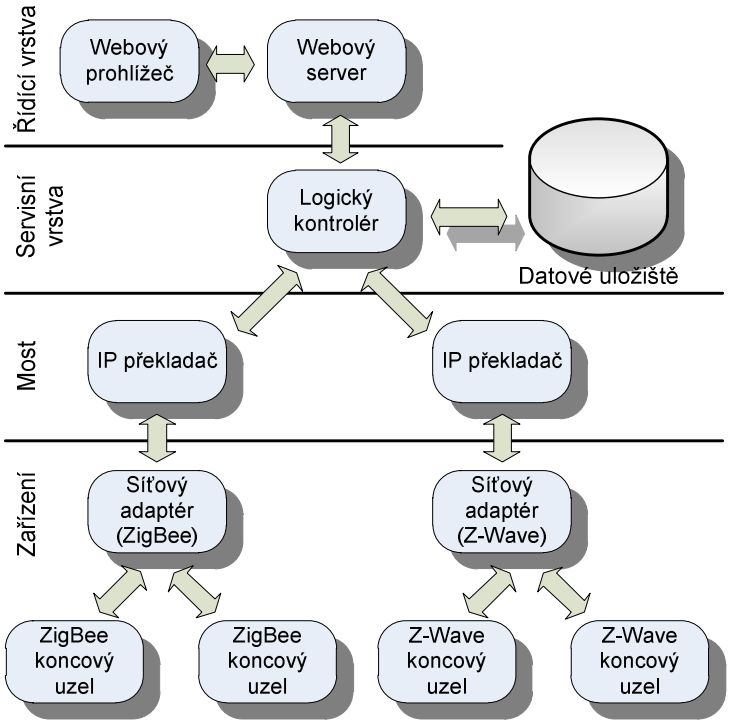
Vrstva zařízení (*Device Layer*), která obsahuje množiny průmyslových senzorů (uzly) některého subsystému jako je ZigBee, BlueTooth nebo Z-Wave. Do této vrstvy patří nejen všechny koncové uzly dané senzorové sítě, ale i síťové směrovače a částečně i hlavní řídící prvek sítě.

Přemosťující vrstva (*Bridge Layer*), je zodpovědná za překlad datových rámců mezi dvěma různými heterogenními síťovými protokoly. Na této vrstvě se nachází malá aplikace, která průběžně transformuje příchozí zprávy z jedné síťové architektury do druhé. V našem případě používáme jako výchozí síťovou architekturu klasický Ethernet. Tzn., že pro každou síťovou architekturu máme malou aplikaci, která transformuje zprávy do TCP/IP. Aplikace *bridge* běží na síťovém zařízení, které má rozhraní do senzorové sítě a Ethernetu. Může se stát, že toto zařízení může být složeno ze dvou hardwarových komponent. První může obsahovat pouze rozhraní do IP sítí, druhé pak rozhraní do senzorové sítě. Touto druhou HW komponentou obvykle bývá koordinátor senzorové sítě. Obě komponenty jsou pak propojeny sériovým rozhraním.

Servisní vrstva (*Service Layer*) je vrstva, která tvoří hlavní funkční blok pro komunikaci se senzorovou sítí. V této vrstvě jsou implementovány základní funkce a procedury, které dokáží složit požadavek a vyvolat jeho provedení na vzdáleném inteligentním senzoru. Na základě tohoto volání se na senzoru operace provede. Servisní vrstva dokáže přijmout zprávu a předá ji k vyhodnocení do řídící vrstvy.

Řídící vrstva (*Management Layer*) obsahuje řídící a rozhodovací funkce, dále algoritmy pro převod a konverzi dat, resp. hodnot mezi jednotlivými koncovými zařízeními. Koncová zařízení se mohou nalézat v různých sítích s různými komunikačními protokoly.

Dekompozicí vrstveného modelu můžeme získat flexibilní komponentově-orientovanou bránu určenou pro multi-doménový řídicí systém, viz Obrázek 12. Jednotlivé komponenty nemusí využívat stejnou síťovou architekturu, ale několik různých síťových standardů a architektur jako je ZigBee, Z-Wave nebo BlueTooth.



Obrázek 12: Komponentový model univerzální brány [24]

Koncové uzly jsou zařízení, na kterých je umístěn senzor nebo více senzorů jako je teplotní čidlo, tlakové čidlo, kouřové čidlo, vypínač, světlo apod. Tento koncový uzel komunikuje skrze senzorovou síť se svým síťovým adaptérem (u ZigBee s koordinátorem), a to buď přímo anebo přes nějaký směrovač.

IP překladač provádí zabalení rámce senzorové sítě do paketu protokolu TCP nebo UDP a posílá skrze Ethernetovou síť (Internet) na řídící logiku.

6.1.3 Profil senzoru

Profil senzoru definuje základní charakteristiky senzoru a je přiřazen ke každému uzlu. Tyto charakteristiky slouží logické komponentě k rozpoznání typu senzoru, ke zjištění seznamu povolených operací, data kalibrace senzoru a spoustu jiných statistických vlastností, které se odvíjejí od aktuální role senzoru. Senzorový profil je nejvhodnější uchovávat v XML formátu, který je lehce upravitelný na základě požadavků aplikace nebo uživatele. Náš profil se skládá ze dvou částí: (i) *AttributeProfile*, který popisuje základní charakteristiky senzoru jako je typ senzoru, umístění, výrobce, přesnost, A/D rozlišení, apod.; (ii) *DataProfile*, který popisuje formát dat generovaných senzorem. Hodnotou může být celé číslo nebo reálné číslo, jedna hodnota nebo více hodnot, resp. se určuje počet bajtů použitých pro reprezentaci dané vlastnosti.

6.1.4 IP překladač

Každý IP překladač je složen ze dvou sdílených pamětí (*buffers*), které slouží pro komunikaci mezi vlákny klientů a vláknem zabezpečujícím komunikaci se sériovým rozhraním síťového adaptéru senzorové sítě. Pro každého klienta připojujícího se k IP překladači se vytváří klientské vlákno, které zpracovává jeho požadavky nad sdílenými pamětmi. Zpráva přijatá klientským vláknem je složená z příkazu a parametrické části. Tyto příkazy lze rozdělit do dvou skupin:

- Požadavek na zaslání datového rámce do senzorové sítě. Jedná se o příkaz *comm*, jehož parametrem je datový rámec, který se má přeposlat do senzorové sítě.
- Ostatní příkazy umožňující vzdálené nastavení IP překladače, autentizaci klienta, apod.

6.1.5 Síťový adaptér

Sériové vlákno přeposílá zprávu přes sériovou linku do síťového adaptéru, který se stará o veškerou komunikaci se senzorovou sítí (ZigBee, Z-Wave nebo BlueTooth) a sběru dat ze senzorové sítě, což z něj dělá hlavní řídící člen senzorové sítě. Síťový adaptér umožňuje obousměrnou komunikaci. Je schopen přijímat příkazy pro senzorovou síť ze strany serveru, zpracovat je a zaslat příslušným senzorům. Na druhé straně umí přijmout informace ze senzorů, vyhodnotit je a zaslat zpět serveru přes IP překladač.

6.1.6 Řídící kontrolér

Řídící kontrolér je tvořen logikou aplikace a servisními voláními. Podle obrázku 12 odpovídá komponentám (i) logického kontroléru, přes který komunikuje se síťovým adaptérem skrze IP překladač a (ii) webovém serveru, který plní funkci řídící aplikace. Řídící kontrolér plní úlohy, kterými je realizovaný monitoring a ovládání senzorů: (i) Přijímá a zpracovává zprávy od síťového adaptéru. Např. událost (*TRAP* zpráva) nebo odpověď na nějaký příkaz (zjistit stav světla nebo aktuální teplotu). (ii) Provádí analýzu přijatých informací a hodnoty zapisuje do XML, které popisuje stav sítě. Přímá pokyny od uživatele a přeposílá je síťovému adaptéru.

6.1.6.1 Zpracování dotazů - konvertor

Každý senzor připojený do tohoto systému dostane svůj speciální identifikátor *logicSensorID*. Nad tímto identifikátorem jsou prováděny operace, které jsou mapovány na fyzické uzly, resp. senzory. Po přijetí příkazu webovým serverem dochází k jeho zpracování ve vstupním modulu (*Input Module*). V rámci tohoto modulu může docházet k tzv. URL směrování (*URL routing*), což je transformace URL z jednoho tvaru do druhého. Např. příkaz pro získání hodnoty *GET /single?logicSensorID=1&get* může být transformován do tvaru *GET /logicSensorID/1/get*, nebo pokud chceme adresovat skupinu uzlů, použijeme dotaz ve tvaru *GET /single?group=1&set=2*, resp. *GET /logicGroupID/1/set/2*. Obecnou syntaxi požadavku je možné definovat jako:

GET / logicSensorID / <id_sensoru> / <nazev_operace> / <id_operace> / <volitelý_parametr_1> / <volitelný_parametr_2> /… / <volitelný parametr n>

Ve druhé fázi dojde k extrakci parametrů a nalezení logického senzoru a jeho operace v seznamu operací. Po nalezení operace se identifikuje senzorová a síť a vytvoří se požadavek pro síťový adaptér. Zpráva se odesílá na IP překladač. Ukázka 4 (str. 25) zobrazuje definici senzoru v logickém kontroléru.

Paket přenášený ke koordinátoru senzorové sítě je složen z:

- 2 bajtového pole označující délku paketu
- sekvencí dalších atributů uvedených v sekci *operation* vztahující se k dané operaci, viz Ukázka 1:
    - 16bitové adresy cílového uzlu
    - hlavičky APS rámce
    - adresními atributy
    - datovou částí: ZCL příkazem nebo libovolným obsahem

```
<sensors>
  <!-- identifikace senzoru -->
  <sensor id='1' name='teplota' type='temperature'>
    <description>teplotní čidlo</description>
    <network_id id='1'>ZigBee1</network_id>
    <!-- nětnitelné hodnoty senzoru -->
    <values>
      <value id='1' name='teplota' operation_id='1' param='1'
             type='uint16'>27</value>
    </values>
    <!-- seznam operací -->
    <operation_list>
      <operation id='1' name='get' type='request’>
        <dest_address='uint16'>0x669F</dest_address>
        <aps_frame_control type='uint8'>0x00</aps_frame_control>
        <destination_endpoint type='uint8'>0xE8</destination_endpoint>
        <cluster_id type='uint16'>0x0500</cluster_id>
        <profile_id type='uint16'>0xE8C1</profile_id>
        <source_endpoint type='uint8'>0xE1</source_endpoint>
        <data type=’uint8'>0x04</data>
      </operation>
    </operation_list>
  </sensor>
```

Ukázka 1: Virtuální senzor – syntaxe požadavků

Po přijetí zprávy koordinátor doplní cílovou adresu uzlu, APS hlavičku, sekvenční číslo rámce (doplněno koordinátorem) a poté do bufferu uloží datový obsah se ZCL příkazem nebo jinou zprávou. Poté je paket odeslán do senzorové sítě.

Podobným způsobem bude zpracována odpověď, která přijde ze strany koordinátoru. Na základě adresy cílového uzlu a adresních atributů aplikace budou nalezeny odpovídající senzory a identifikuje se operace. Při identifikaci operace se zjistí, jakým způsobem se bude mapovat obsah datové části na atributy *value*. Ukázka 5 zobrazuje definici snímače, který má nadefinovanou operaci pro přijetí odpovědi s datovou částí o velikosti 16bitů, jehož prvních 8 bitů bude uloženo do první proměnné a druhých 8bitů do druhé proměnné.

```
<sensors>
  <sensor id='1' name='teplota' type='temperature'>
    <description>teplotní čidlo</description>
    <network_id id='1'>ZigBee1</network_id>
    <values>
      <value id='1' name='teplota' operation_id='1' param='1'
             type='uint16'>27</value>
      <value id='1' name='teplota' operation_id='1' param='1'
             type='uint16'>27</value>
    </values>
    <operation_list>
      <operation id='1' name='get' type='response'>
        <dest_address='uint16'>0x0000</dest_address>
        <aps_frame_control type='uint8'>0x00</aps_frame_control>
        <destination_endpoint type='uint8'>0xE8</destination_endpoint>
        <cluster_id type='uint16'>0x0500</cluster_id>
        <profile_id type='uint16'>0xE8C1</profile_id>
        <source_endpoint type='uint8'>0xE1</source_endpoint>
        <param id='1' type='uint8' conversion='./module/conv/conv1.py'
               value_id='1'/>
        <param id='2' type='uint8' conversion='./module/conv/conv1.py'
               value_id='1'/>
      </operation>
    </operation_list>
  </sensor>
</sensors>
```

Ukázka 2: Virtuální senzor – syntaxe odpovědi

6.2 Shrnutí

Při připojení nových typů zařízení je možné provádět pouze aktualizaci logického kontroléru ve smyslu úpravy zdrojových XML kódů, kdy se nadefinuje nový typ senzoru a sekvence a typ jednotlivých atributů. Takto poslaný rámec se pošle na síťový adapter a koordinátor sítě na jeho základě vyplní fyzický rámec, který se odešle k cílovému uzlu. Podobným způsobem je zmapována i příchozí odpověď nebo náhodná událost. Podobným způsobem je možné realizovat připojení sítí typu BlueTooth nebo Z-Wave.

V případě, že by se komunikační protokol mezi logickým kontrolérem a síťovým zařízením definoval na úrovni celého APS rámce včetně ZLC rámce s přídavkem zdrojové a cílové adresy uzlu, přenesla by se tak plná funkčnost ZigBee koordinátoru do logického kontroléru. Jinými slovy, koordinátor by plnil jen funkce pro síťovou vrstvu, tzn. směrování, autentizace, výměny bezpečnostních klíčů a výpočet kontrolních součtů. Veškerá aplikační část by se nacházela v logickém kontroléru. Návrh definice APS rámce a ZLC rámce je možné najít v příloze A a B.

Dalším přínosem tohoto řešení je možnost zavedení redundance řídícího kontroléru (logický kontrolér a serverová část) a to tak, že si redundantní kontrolér vytvoří nové spojení s IP překladačem a svoje data zesynchronizuje s primárním kontrolérem.

Při spuštění sekundárního logického kontroléru se provede synchronizace s primárním kontrolérem a připojí se k jednotlivým IP překladačům podle schématu na primárním. Sekundární kontrolér periodicky kontroluje primární kontrolér a v případě zjištění výpadku dojde k přenesení funkčnosti na sekundární. Redundantní kontrolér může sloužit jako monitorací centrum, které může provádět analýzu neznámých zpráv a sloužit tak jako vývojové prostředí pro technika, který přidává do sítě nové zařízení a testuje jeho funkčnost.

7 Závěr

Dizertační práce pojednává o problematice v oblasti bezdrátových senzorových sítí pro vestavěné systémy a koncepci prostředí pro jejich využití při návrhu řešení systému. V počátečních kapitolách jsou probírány popisy standardních řešení týkající se návrhu inteligentního snímače ve smyslu modulárního systému složeného z komponent zajišťujících požadovanou činnost. Jedním z těchto modulů je modul síťového rozhraní pro komunikaci s ostatními zařízeními v síti. Poté následuje kapitola popisující síťové architektury, které jsou vhodné pro tvorbu vestavěných aplikací komunikujících v bezdrátovém prostředí WPAN. Z bezdrátových síťových architektur pro vestavěné aplikace byla vybrána síť typu ZigBee z důvodu její energetické nenáročnosti, otevřené specifikace, komplexnosti a nedávnému přijetí jako standardu pro výrobce podporující tuto technologii. V rámci experimentu bylo otestováno chování modifikovaného směrovacího algoritmu AODV, u kterého se omezuje všesměrové vysílání paketů kontrolujících lokální konektivitu sousedů. Jako jedna z dalších variant byla představena myšlenka směrování založená na základě spotřebované energie při zaslání zprávy sousedovi.

V práci byly dále představeny případové studie jako návrhové vzory pro připojení senzorové sítě k síti Internetu dvěma způsoby a to (i) pomocí vlastního jazyka a (ii) pomocí protokolu SNMP. Na základě těchto vzorů byla navržena architektura univerzální brány pro vestavěné aplikace pracující s různými senzorovými sítěmi. Díky tomuto řešení je možné komunikovat s koncovými uzly napříč síťovými architekturami a zároveň je možné přenést řídící logiku mimo senzorovou síť. Dále navržený systém umožňuje připojení redundantního řídícího kontroléru, který je schopen převzít řízení v případě výpadku primárního kontroléru.

Literatura

[1] 1451.2-1997, I. S. *Standard for a Smart Transducer Interface for Sensors and Actuators - Transducer to Microprocessor* …. New Jersey: Institute of Electrical and Electronics Engineers, Inc. 1997.

[2] California Eastern Laboratories. *ZIC2410 User's Guide Profile and ZigBee Cluster Library (ZCL) for the CEL ZigBee Stack 0005-05-08-04-001*, 2005.

[3] Callaway E., Demuth B., Znati T. *Wireless sensor networks*: *an interdisciplinary approach to designing fast networked devices*. Vyd. 1. London: CRC Press, 2004, 342 s. ISBN 08-493-1823-8.

[4] Koval M. *Technologie senzorových sítí*. Diplomová práce, FIT VUT Brno, CZ, 2006

[5] Jánský V. *Směrování v bezdrátových sítích*. Diplomová práce, FIT VUT Brno, CZ, 2007

[6] *IEEE 1451.1: Standard for a Smart Transducer Interface for Sensors and Actuators - Network Capable Application Processor (NCAP) Information Model*, IEEE, New York, 2000.

[7] *IEEE 1451.2: Standard for a Smart Transducer Inter-face for Sensors and Actuators - Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats*, IEEE, New York, 1997.

[8] Ilyas, M., Mahgoub, I. *Wireless Internet Handbook: Technologies, Standards, and Applications*. Boca Raton, CRC Press LLC, 2003, ISBN: 0-8493-1502-6.

[9] Jurdak R., Lopes C. V., Baldi P. *A framework for modeling sensor networks*. In *Workshop on Building Software for Pervasive Computing*, USA, 2004

[10] Keshav S. *An Engineering Approach to Computer Networking*, Massachusetts: Addison-Wesley, 1997, 660 s. ISBN 02-016-3442-2.

[11] Lattibeaudiere D. P. *AnI232 - Microchip ZigBee-2006 Residential Stack Protocol*. Microchiop. 2008.

[12] Perman, P. *Dálkové řízení modelu*, Diplomová práce, České Vysoké učení technické v Praze, 2005.

[13] Pužmanová R.: *Bezpečnost bezdrátové komunikace: jak zabezpečit wi-fi, BlueTooth, GPRS či 3G*. Brno: Computer Press, 2005, ISBN: 80-251-0791-4.

[14] Sedlák F. *Systém pro sledování a správu senzorových sítí*, Diplomová práce, FIT VUT Brno, CZ, 2007

[15] Shorey, R. *Mobile, Wireless, and Sensor Networks: Technology, Applications, Future Directions*. Vyd. 1. New Jersey: John Wiley, 2006, 430 s. ISBN 04-717-1816-5.

[16] Švéda, M. Routers and Bridges for Small Area Network Interconnection, In *Computers in Industry, Vol.22, No.1*, Elsevier Science, Amsterdam, NL, 1993, pp.25 29.

[17] Švéda, M., Trchalík, R. Development of Interconnecting SW for Intranets and Fieldbuses, In *10th IFAC Workshop on Programmable Devices and Embedded Systems*, PDeS 2010, Pszczyna, PL, IFAC, 2010, p. 119-124, ISSN 1474-6670.

[18] Švéda, M., Trchalík, R. Safety and Security-driven Design of Networked Embedded Systems, In *Proceedings 10th EUROMICRO Conference on Digital System Design Architectures, Methods and Tools*, Lübeck, DE, IEEE CS, 2007, p. 420-423, ISBN 0-7695-2978-X.

[19] Švéda, M., Trchalík, R. Smart Sensor Networking with ZigBee and Internet, In *Proceedings of the 2nd International Workshop on Artificial Neural Networks and Intelligent Information Processing (ANNIIP 2006)*, Setúbal, PT, UPMC, 2006, p. 64-71, ISBN 972-8865-68-6.

[20] Švéda, M., Trchalík, R. ZigBee-to-Internet Interconnection Architectures, In *Proceedings of the Second International Workshop on Mobile Communications and Learning MCL 2007*, Saint Luce, Martinique, MQ, IEEE CS, 2007, p. 6, ISBN 0-7695-2807-4.

[21] Trchalík, R., Očenášek, P. Addressing and Routing in Sensor Networks, In *Proceedings of the 8th International Carpathian Control Conference*, Košice, SK, TU v Košiciach, 2007, p. 4, ISBN 978-80-8073-805-1.

[22] Trchalík, R., Švéda, M. Sensor Networking through Intranet and ZigBee, In *EDS '07 IMAPS CS International Conference Proceedings*, Brno, CZ, VUT v Brně, 2007, p. 1-5, ISBN 978-80-214-3470-7.

[23] Trchalík, R., Švéda, M. Sensor Networking through Intranet and ZigBee, In *Proceedings IMAPS CS International Conference EDS'06*, Brno, CZ, VUT v Brně, 2006, p. 217-221, ISBN 80-214-3246-2.

[24] Trchalík, R., Švéda, M. Unified Sensor Gateway Interconnection of Sensor Networks, In *Proceedings 11th IFAC/IEEE International Conference on Programmable Devices and Embedded Systems*, Brno, CZ, FEKT VUT, 2012, p. 348-353, ISSN 1474-6670.

[25] Varghese G., Minoli D., Znati T. *Network algorithmics*: *an interdisciplinary approach to designing fast networked devices*. Vyd. 1. Amsterdam: Elsevier, c2005, xxiv, 465 s. ISBN 01-208-8477-1.
[26] Walls C., Minoli D., Znati T. *Embedded software: the works*. Vyd. 1. Burlington: Newnes, 2006, xxiv, 390 s. ISBN 07-506-7954-9.
[27] Z-Wave Alliance, Z-Wave Z-Wave Node Type Overview and Network Installation Guide. INS10244. 2008.
[28] ZigBee Alliance, *ZigBee Cluster Library Specification*. Dokument 075123r02ZB. 2008.
[29] ZigBee Alliance. *ZigBee Specification v 1.0*. Dokument 053473r00. 2004.
[30] ZigBee Alliance, *ZigBee Specification*. Dokument 053474r17. 2008.

Internetové zdroje:

[31] Brief Description of the Family of IEEE 1451 Standards [cit. 2012-02-25]. Dostupné na URL: < http://www.nist.gov/el/isd/ieee/1451family.cfm>
[32] RFC 1155: Structure and Identification of Management Information for the TCP/IP-based Internets. [cit. 2008-05-04]. Dostupné na URL: < http://tools.ietf.org/html/rfc1155>
[33] RFC 1212: Concise MIB Definitions. [cit. 2008-05-04]. Dostupné na URL: <http://tools.ietf.org/html/rfc1212>
[34] RFC 1213: Management Information Base for Network Management of TCP/IP-based internets: MIB-II. [cit. 2008-05-04]. Dostupné na URL: < http://tools.ietf.org/html/rfc1213>
[35] RFC 1215: A Convention for Defining Traps for use with the SNMP [cit. 2008-05-04]. Dostupné na URL: <http://tools.ietf.org/html/rfc1215>
[36] RFC 3561: Ad hoc On-Demand Distance Vector (AODV) Routing [cit. 2006-05-04]. Dostupné na URL: <http://tools.ietf.org/html/rfc3561>
[37] UART, Universal asynchronous receiver/transmitter. [cit. 2009-04-20]. Dostupné na URL: http://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter

Přehled publikační činnosti

[1] Očenášek, P., Trchalík, R., Švéda, M.: On the Implementation of Metrics in Industrial Embedded Systems, In: Preprints of IFAC Workshop on PROGRAMMABLE DEVICES and EMBEDDED SYSTEMS PDeS 2009, Ostrava, CZ, IFAC, 2009, p. 161-164, ISSN 1474-6670.

[2] Očenášek, P., Trchalík, R.: Approaches to the Security and Payment Protocols Design: State of the Art, In: *International Conference on INFORMATION TECHNOLOGY INTERFACES*, Dubrovnik, HR, IEEE, 2011, p. 4, ISBN 978-953-7138-21-9

[3] Očenášek, P., Trchalík, R.: Modal Logics Used for Authentication Protocols Analysis: Survey and Comparison, In: *Proceedings of the 7th International Carpathian Control Conference*, Ostrava, CZ, VŠB TU, 2006, p. 401-404, ISBN 80-248-1066-2.

[4] Očenášek, P., Trchalík, R.: On the Implementation of Metrics in the Workflow System, In: *Proceedings of the 6th International Conference on Applied Computer Science*, Puerto De La Cruz, ES, WSEAS, 2006, p. 329-331, ISBN 960-8457-57-2.

[5] Očenášek, P., Trchalík, R.: On the Implementation of Metrics in the Workflow System, In: *WSEAS Transactions on Computers Research, Vol. 1, No. 2*, 2006, Athens, GR, p. 360-362, ISSN 1991-8755.

[6] Očenášek, P., Trchalík, R.: Reasoning About Security Protocols in the ZigBee Standard, In: *Proceedings of the 8th International Carpathian Control Conference*, Košice, SK, TU v Košiciach, 2007, p. 4, ISBN 978-80-8073-805-1.

[7] Očenášek, P., Trchalík, R.: The Use of Modal Logics in the Security Protocols Analysis, In: *Proceedings of the 12th Conference STUDENT EEICT 2006*, Brno, CZ, FEKT VUT, 2006, p. 395-399, ISBN 80-214-3163-6.

[8] Očenášek, P., Trchalík, R.: Tracing Authentication Protocols Behavior: A Case Study, In: *MEMICS 2006 Second Doctoral Workshop on Mathematical and Engineering Methods in Computer Science*, Mikulov, CZ, FIT VUT, 2006, p. 166-173, ISBN 80-214-3287-X.

[9] Očenášek, P., Trchalík, R.: Using AVISPA in the Education of Network Security, In: *EDS '08 IMAPS CS International Conference Proceedings, Brno*, CZ, VUT v Brně, 2008, p. 8, ISBN 978-80-214-3717-3.

[10] Švéda, M., Trchalík, R., Očenášek, P.: Design of Networked Embedded Systems: An Approach for Safety and Security, In: *Preprints of IFAC Workshop on PROGRAMMABLE DEVICES and EMBEDDED SYSTEMS PDeS 2009*, Ostrava, CZ, IFAC, 2009, p. 131-136, ISSN 1474-6670.

[11] Švéda, M., Trchalík, R.: Development of Interconnecting SW for Intranets and Fieldbuses, In: *10th IFAC Workshop on Programmable Devices and Embedded Systems*, PDeS 2010, Pszczyna, PL, IFAC, 2010, p. 119-124, ISSN 1474-6670.

[12] Švéda, M., Trchalík, R.: Safety and Security-driven Design of Networked Embedded Systems, In: *Proceedings 10th EUROMICRO Conference on Digital System Design Architectures, Methods and Tools*, Lübeck, DE, IEEE CS, 2007, p. 420-423, ISBN 0-7695-2978-X.

[13] Švéda, M., Trchalík, R.: Smart Sensor Networking with ZigBee and Internet, In: *Proceedings of the 2nd International Workshop on Artificial Neural Networks and Intelligent Information Processing (ANNIIP 2006)*, Setúbal, PT, UPMC, 2006, p. 64-71, ISBN 972-8865-68-6.

[14] Švéda, M., Trchalík, R.: ZigBee-to-Internet Interconnection Architectures, In: *Proceedings of the Second International Workshop on Mobile Communications and Learning MCL 2007*, Saint Luce, Martinique, MQ, IEEE CS, 2007, p. 6, ISBN 0-7695-2807-4.

[15] Trchalík, R., Očenášek, P., Švéda, M.: Using MSC and SDL Languages for Description of Network Communication, In: *EDS '08 IMAPS CS International Conference Proceedings*, Brno, CZ, VUT v Brně, 2008, p. 5, ISBN 978-80-214-3717-3.

[16] Trchalík, R., Očenášek, P.: Addressing and Routing in Sensor Networks, In: *Proceedings of the 8th International Carpathian Control Conference*, Košice, SK, TU v Košiciach, 2007, p. 4, ISBN 978-80-8073-805-1.

[17] Trchalík, R., Očenášek, P.: Metrics in Workflow Systems, In: *Proceedings of the 7th International Carpathian Control Conference*, Ostrava, CZ, VŠB TU, 2006, p. 569-572, ISBN 80-248-1066-2.

[18] Trchalík, R., Očenášek, P.: ZigBee Gateways, In: *Proceedings of the 12th Conference STUDENT EEICT 2006*, Brno, CZ, FEKT VUT, 2006, p. 410-414, ISBN 80-214-3163-6.

[19] Trchalík, R., Švéda, M.: Sensor Networking through Intranet and ZigBee, In: *EDS '07 IMAPS CS International Conference Proceedings*, Brno, CZ, VUT v Brně, 2007, p. 1-5, ISBN 978-80-214-3470-7.

[20] Trchalík, R., Švéda, M.: Sensor Networking through Intranet and ZigBee, In: *Proceedings IMAPS CS International Conference EDS'06*, Brno, CZ, VUT v Brně, 2006, p. 217-221, ISBN 80-214-3246-2.

[21] Trchalík, R., Švéda, M.: Unified Sensor Gateway Interconnection of Sensor Networks, In: *Proceedings 11th IFAC/IEEE International Conference on Programmable Devices and Embedded Systems*, Brno, CZ, FEKT VUT, 2012, p. 348-353, ISSN 1474-6670.

[22] Trchalík, R.: A Component-Based model for Embedded Software, In: *EDS '09 IMAPS CS International Conference Proceedings*, Brno, CZ, CZ, VUT v Brně, 2009, p. 5, ISBN 978-80-214-3933-7.

[23] Trchalík, R.: Design IEEE 802.15.4 ZigBee sítě, In: *Počítačové architektúry & diagnostika 2006*, Bratislava, SK, SAV, 2006, p. 107-112, ISBN 80-969202-2-7.