

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

APLIKACE PRO ADAPTABILNÍ DEKÓDOVÁNÍ PAKETŮ V
BEZDRÁTOVÉ SÍTI

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

VOJTĚCH PISKA

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

APLIKACE PRO ADAPTABILNÍ DEKÓDOVÁNÍ PAKETŮ V BEZDRÁTOVÉ SÍTI

APPLICATION FOR ADAPTIBLE PACKET DECODING IN WIRELESS NETWORKS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

VOJTĚCH PISKA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MILAN ŠIMEK, Ph.D.

BRNO 2013



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Vojtěch Piska

ID: 136039

Ročník: 3

Akademický rok: 2012/2013

NÁZEV TÉMATU:

Aplikace pro adaptabilní dekódování paketů v bezdrátové síti

POKYNY PRO VYPRACOVÁNÍ:

Student v rámci bakalářské práce navrhne a realizuje aplikaci, která bude sloužit pro dekódování uživatelsky zadaných protokolů pro bezdrátové senzorové sítě. Úkolem studenta je navrhnout a naprogramovat webovou aplikaci, tj. vytvořit modul pro definici nových protokolů, pro dekódování příchozích paketů a dále také vytvořit modul pro grafickou reprezentaci naměřených dat. Součástí projektu je i návrh grafického rozhraní této aplikace.

DOPORUČENÁ LITERATURA:

- [1] Stojmenovic I., Handbook of Sensor Networks, Wiley, ISBN:13 978-0-471-68472-5, 2005.
- [2] FARAHANI, Shahin. Zigbee Wireless Networks and Transceivers. [s.l.] : Elsevier, 2008. 329 s. ISBN

Termín zadání: 11.2.2013

Termín odevzdání: 5.6.2013

Vedoucí práce: Ing. Milan Šimek, Ph.D.

Konzultanti bakalářské práce:

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

V této bakalářské práci jsou popsány základní technologie pro vývoj webových aplikací. Je zde popsána struktura aplikace pro dekódování paketů a podrobné fungování jejich součástí.

KLÍČOVÁ SLOVA

PHP, MYSQL, JavaScript, dekódování, paket

ABSTRACT

In this bachelor thesis are described basic technics for developing web applications. Thesis contains structure of application for packet decoding. There is detailed description of function each part.

KEYWORDS

PHP, MYSQL, JavaScript, decoding, packet

PISKA, Vojtěch *Aplikace pro adaptabilní dekódování paketů v bezdrátové síti*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2013. 36 s. Vedoucí práce byl Ing. Milan Šimek, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Aplikace pro adaptabilní dekódování paketů v bezdrátové síti“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Milanu Šimkovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Výzkum popsáný v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

.....
(podpis autora)

OBSAH

Úvod	9
1 Požadavky na aplikaci	10
2 Struktura rámce v bezdrátových senzorových sítích	11
3 Použité technologie	12
3.1 PHP	12
3.2 MySQL	12
3.3 JavaScript	13
3.3.1 jQuery	15
3.4 CSS	15
3.5 HighStock	16
4 Praktická realizace	17
4.1 Struktura aplikace	17
4.1.1 Hlavní modul aplikace	18
4.1.2 Modul definice protokolu	19
4.1.3 Validace	21
4.1.4 Modul generování dat	23
4.1.5 Modul dekódování dat	25
4.1.6 Grafický modul	26
4.1.7 Databáze	28
5 Závěr	31
Literatura	32
Seznam symbolů, veličin a zkratk	34
Seznam příloh	35
A médium CD	36

SEZNAM OBRÁZKŮ

2.1	Struktura MAC rámce a složení kontrolního pole	11
3.1	Schéma fungování PHP a MySQL	13
3.2	Schéma fungování JavaScriptu	14
4.1	Struktura aplikace	17
4.2	Hlavní okno aplikace	19
4.3	Definice nového protokolu	21
4.4	Oznámení o tom, že byl protokol uložen do databáze	21
4.5	Chybové hlášení v případě nevyplnění vstupního pole	22
4.6	Dekódovaná data	25
4.7	Grafický modul	27
4.8	Obsah tabulky data	30
4.9	Obsah tabulky prot	30

ÚVOD

V této bakalářské práci byla navržena a následně realizována adaptabilní webová aplikace sloužící k dekodování příchozích paketů.

Zadání struktury vlastního protokolu a její následné užití při zpracování dat podporuje i velmi oblíbený protokolový analyzátor Wireshark. Vytváření pluginů (dissectorů ve Wiresharku), které umožňují analyzovat vlastní protokoly, je však zdlouhavé a pracné. Proto bylo cílem vytvořit jednodušší strukturu, která by však bohatě dostávala potřebám uživatelů.

V první kapitole jsou charakterizovány použité technologie, na kterých byla celá aplikace vystavěna. V následujících kapitolách je již popsána realizace samotné aplikace a součástí s ní související. Samotná aplikace se skládá z dílčích modulů, které je možné dále rozšiřovat, přidávat, a tak samotnou aplikaci dále rozvíjet. Je tak zajištěna určitá modularita.

Cílem práce bylo vytvořit modul pro zadávání nových protokolů, u kterých si mohou sami uživatelé definovat vlastní strukturu, podle které se pak budou příchozí data dekodovat. Další modul slouží ke generování simulovaných dat do databáze. Tyto simulované data představují reálné příchozí pakety. Dále se zde nachází modul, který slouží k dekodování příchozích dat a modul pro grafickou vizualizaci dat.

1 POŽADAVKY NA APLIKACI

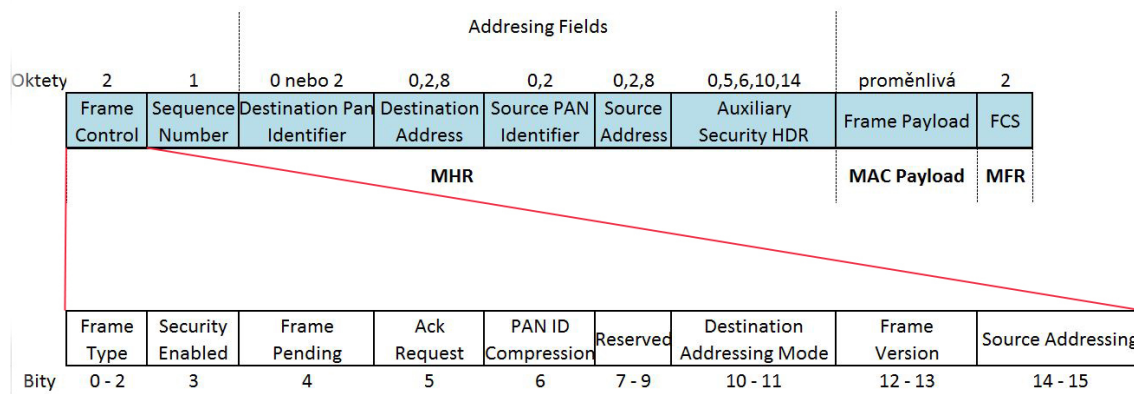
Při tvorbě aplikace byla snaha vytvořit jednoduchou, srozumitelnou strukturu, která by byla uživatelsky přívětivá a lehce ovladatelná. Požadavkem byla webová aplikace, ke které by se dalo přistupovat z internetového prohlížeče a používat ji prakticky odkudkoliv. Cílem bylo vytvoření aplikace tak, aby její struktura byla modulární a snadno rozšiřitelná o případné další funkce bez narušení její funkčnosti.

Uživatel si může vytvořit vlastní protokol, který se uloží do databáze a ten pak použít pro dekodování dat. Případně si pak vybrat ze seznamu již protokolů nadefinovaných v databázi. K získání dat pro dekodování může uživatel využít generátoru dat, který simuluje příchozí pakety do databáze.

Dekodovaná data jsou uživateli zobrazena v tabulkové struktuře a také v grafické závislosti pro větší přehlednost.

2 STRUKTURA RÁMCE V BEZDRÁTOVÝCH SENZOROVÝCH SÍTÍCH

Pro lepší představu jak vypadá takový rámec, který je vysílán bezdrátovými uzly, si zde ukážeme strukturu MAC rámce (obr.2.1). Jeho maximální velikost je 127 bajtů, jelikož je na fyzické vrstvě zapouzdřen do datové části v poli PHY Payload, která má velikost právě 127 bajtů. Jak je patrné z obrázku MAC rámec je složen ze tří částí:



Obr. 2.1: Struktura MAC rámce a složení kontrolního pole

MHR(Mac Header), MAC Payload a MFR(Mac Footer). Velikosti jednotlivých polí jsou v bajtech. Pole FCS (Frame Control Sequence) definuje o jaký typ rámce se jedná - Beacon, Data, Command, ACK. Pomocná hlavička Auxiliary Header, přenášející hodnotu úrovně zabezpečení a šifrovaný klíč. Pole Destination Addressing Mode a Source Addressing Mode určují zdali je použito 16-bit nebo 64-bit adresování. Velikost adresní části MAC rámce tedy závisí na použitém adresování. Velikost jednoho adresního pole tak může nabývat 0 bajtů (žádná adresa), 2 bajty (16-bit adresa) nebo 6 bajtů (64-bit adresa). Pole Frame Version určuje verzi standardu IEEE 802.15.4 podle které je rámec strukturován. Sekvenční číslo v MAC rámci definuje sekvenční číslo pro Beacon rámec a sekvenční číslo pro Datový rámec. Jeho velikost je 8 bitů. Poslední pole MAC rámce je kontrola rámce tzv. FCS (Frame Control Sequence). Jedná se o 16-bit hodnotu, vypočtenou z obsahu rámce pomocí algoritmu CRC (Cyclic Redundancy Check). [9]

3 POUŽITÉ TECHNOLOGIE

3.1 PHP

PHP nebo-li Hypertext Preprocessor se řadí mezi skriptovací jazyky provádějící se na straně serveru. Je to jazyk speciálně navržený pro užití ve webových stránkách a je licencován jako Open Source. Jeho velkou předností je to, že PHP skript může být zakomponován do struktury běžné HTML (HyperText Markup Language) stránky, který se provede až při zpracování serverem a uživateli je vrácen do internetového prohlížeče až samostatný výstup činnosti. Samotný proces je zobrazen na obr.3.1. PHP podporuje nativně také mnoho druhů databází. Mezi nimi i MySQL (My Structured Query Language), která je použita i v této práci.

Historie PHP

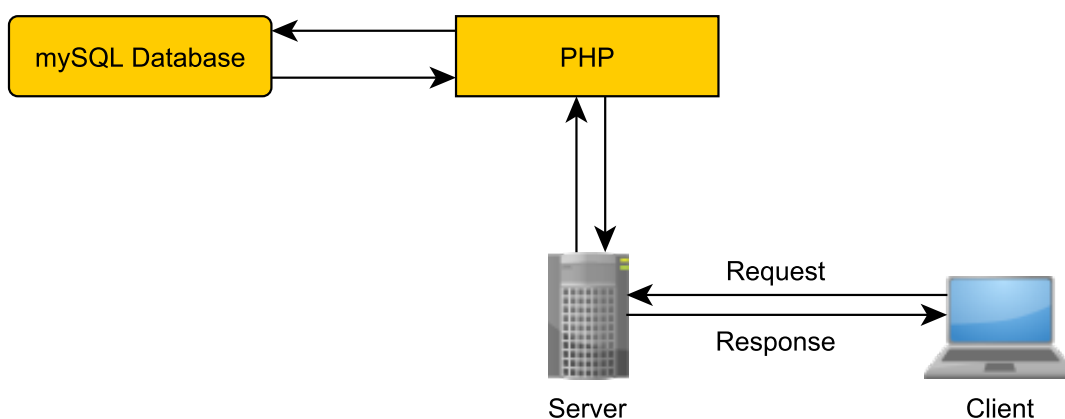
Původní název pro PHP bylo Personal Home Page, čili osobní domácí stránka. Vznik se datuje k roku 1994, kdy dánský programátor Rasmus Lerdorf sepsal kombinaci skriptů v Perlu za účelem údržby své osobní domovské stránky. Tyto skripty například zobrazovaly obsah stránek nebo zaznamenávaly údaje o návštěvnosti jeho stránek. Po spojení Lerdorfových skriptů s Form Interpreterem vznikla kombinace PHP/FI, která již mohla komunikovat s databázemi a tím tak položila základ vzniku dynamických webových aplikací. Veřejná verze PHP byla vydána 8. června 1995, aby bylo kód možno zdokonalit nalezením chyb. Další verze PHP zdokonalovaly verze předchozí a přidávaly nové funkcionality až do aktuální verze PHP5, která obsahuje podporu pro objektové programování.

Přednosti jazyka PHP:

- Specializovaný jazyk pro webové stránky.
- Podpora velkého množství databázových systémů.
- Multiplatformní prostředí.
- Podpora hostingových služeb – PHP se stal webovým standartem.
- Open Source licence a bohatá dokumentace.

3.2 MySQL

MySQL patří mezi relační multiplatformní databáze. Pro svou funkci využívá celosvětově používaného dotazovacího jazyku SQL (Structured Query Language). Systém byl vytvořen švédskou firmou MySQL AB, nyní vlastněný společností Sun Microsystems, která je dceřinou firmou společnosti Oracle Corporation. Databáze



Obr. 3.1: Schéma fungování PHP a MySQL

umožňuje efektivně hledat, ukládat, třídit a seřazovat získaná data. MySQL je distribuováno jak pod bezplatnou licenci GPL (General Public License), tak pod komerční placenou. Nespornou výhodou, která je využita i při tvorbě této aplikace, je komunikace s PHP a práce s jeho proměnnými.

MySQL databáze jsou oblíbené pro svou jednoduchou implementaci a možnost fungovat na více platformách (Linux, Microsoft Windows i jiných). Systém byl optimalizován především pro rychlé provádění SQL dotazů a to někdy i za cenu větších zjednodušení. Například jsou zde jen relativně jednoduché způsoby zálohování. Každá databáze v prostředí MySQL je tvořena z jedné či více tabulek, které mají řádky a sloupce. Jednotlivé tabulky mohou být navzájem provázány a tím se vyhnout redundanci dat. Do řádků se ukládají jednotlivé záznamy. Sloupce pak obsahují název a datový typ jednotlivých polí záznamu.

Přednosti MySQL databází:

- Rychlost a stabilita těchto databází.
- Multiplatformní systém.
- Dvojí typ licencí, zdarma pro nekomerční použití.
- Obrovská rozšířenost a tím i vyplývající podpora a komunita uživatelů.

3.3 JavaScript

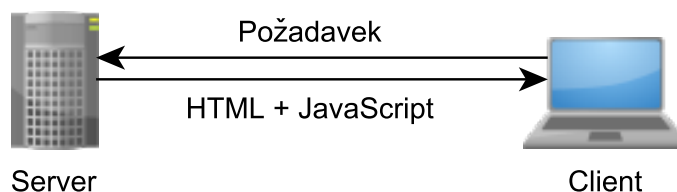
JavaScript je skriptovací multiplatformní jazyk, který se používá ve webových aplikacích. Jedná se o skripty spouštějící se na straně klienta narozdíl od PHP, které se provádí na straně serveru a k uživateli už jsou zasílány výsledky (obr.3.2).

Kód skriptu se často vkládá přímo do HTML těla stránky, popřípadě je k dané stránce připojen externě. JavaScriptem lze v mnoha případech uživatelsky zatraktivnit webové stránky. Může totiž obsluhovat různé prvky na stránce jako jsou tlačítka, textová pole či rolovací seznamy aniž by byla celá stránka znovu načtena. To je užitečné v případě, že potřebujeme například kontrolovat, zda je textové pole před jeho odesláním na server vyplněno.

Jistou nevýhodou této technologie je absence manipulovat se soubory. Ať už na straně klienta či serveru vyjma cookies, s kterými JavaScript pracovat může. Další nevýhodou pak může být fakt, že jej může uživatel v prohlížeči zakázat.

Přednosti JavaScriptu:

- Jde o jazyk skriptovací, tudíž se nemusí kompilovat.
- Provádí se klientově straně, nezatěžuje tak server.
- Umožňuje vykonávat určité funkce kontroly dříve, než se data odešlou na server a jsou zkontrolovány tam.



Obr. 3.2: Schéma fungování JavaScriptu

3.3.1 jQuery

Jedná se o knihovnu, možno označit také jako framework, která usnadňuje ve velké míře práci s JavaScriptem. Její největší předností je práce s elementy DOM (Document Object Model) a možností je modifikovat. Dále pak práce s událostmi a různorodé animační efekty.

Použití je velice snadné. Stačí stáhnout volně dostupnou knihovnu s příponou `jquery.js` ze stránky www.jquery.com. Tu následně naimportovat do adresáře, odkud se na něj odkazovat z hlavičky stránek příkazem:

```
<script language="javascript" type="text/javascript" src="jquery.js"/>
```

Nyní už jen stačí začít využívat možností jQuery. Důležité je umisťovat vytvářené externích skripty, které nejsou umístěny přímo v HTML kódu stránky, do stejného adresáře, kde se nachází soubor `jquery.js`.

3.4 CSS

CSS (Cascading Style Sheets) nebo-li česky kaskádové styly je jazyk sloužící k formátování a zobrazení stránek v jazycích jako jsou HTML, XHTML (Extensive Hyper-Text Markup Language) či XML (Extensible Markup Language). Hlavní výhodou této technologie je možnost oddělit strukturu dokumentu od jejího vzhledu. HTML sice v základu obsahuje jednoduché formátovací příkazy, ovšem jen ve značně jednoduché a omezené míře.

Definice kaskádových stylů sestává z několika pravidel. Každé pravidlo obsahuje selektor a blok deklarací. Každý blok deklarací pak obsahuje seznam deklarací oddělených středníky ; a každá deklarace sestává z identifikátoru vlastnosti, následuje dvojtečka : a hodnota vlastnosti.

Pokud bychom chtěli například upravit zarovnání písma veškerých odstavců na střed pomocí HTML, museli bychom do každého odstavce ohraničenými tagy `<p>...</p>` vkládat parametr `align="center"`. Výsledný příkaz by pak vypadal takto:

`<p align="center">...</p>`. To je ovšem při velkém počtu odstavců nepohodlné. CSS však tento problém řeší elegantněji. V externím souboru s definicemi kaskádových stylů a příponou `*.css` se nadeklaruje:

```
p {  
  align: center;  
}
```


Pak již stačí zahrnout odkaz na soubor `*.css` do hlavičky stránky příkazem:

```
<link rel='stylesheet' type='text/css' href='*.css' />
```

Následně jsou všechny elementy odstavců formátovány podle stylu, který je deklarován v externím souboru.

Přednosti CSS:

- Rozsáhlé možnosti formátování obsahu.
- Jednoduchá údržba webové stránky. Změna vlastnosti konkrétního elementu probíhá jen na jednom místě a následně je aplikována na elementy v celém dokumentu.
- Vlastnosti kaskádových stylů jednotlivých elementů lze dynamicky měnit pomocí JavaScriptu.

3.5 HighStock

Knihovna HighStock je volně šiřitelná knihovna, která umožňuje vykreslování interaktivních grafů na webových stránkách. Tato knihovna je volně dostupná pro nekomerční použití. V jiném případě je třeba držet se licenčních podmínek. Je možné ji stáhnout na adrese www.highcharts.com.

Velkou předností této knihovny je fakt, že je dostupný její kompletní zdrojový kód a je možné v něm provádět vlastní úpravy. Vše, co je potřeba stáhnout, jsou pouze dva JavaScriptové soubory `highstock.js` a `jquery.js`. Ty následně vložit do hlavičky stránky. Na klientově straně tak není potřeba instalovat žádné Flash či Java plugíny. Další výhodou jsou obrovské možnosti nastavení ve srovnání s podobnými knihovnami, přičemž nastavení je poměrně jednoduché.

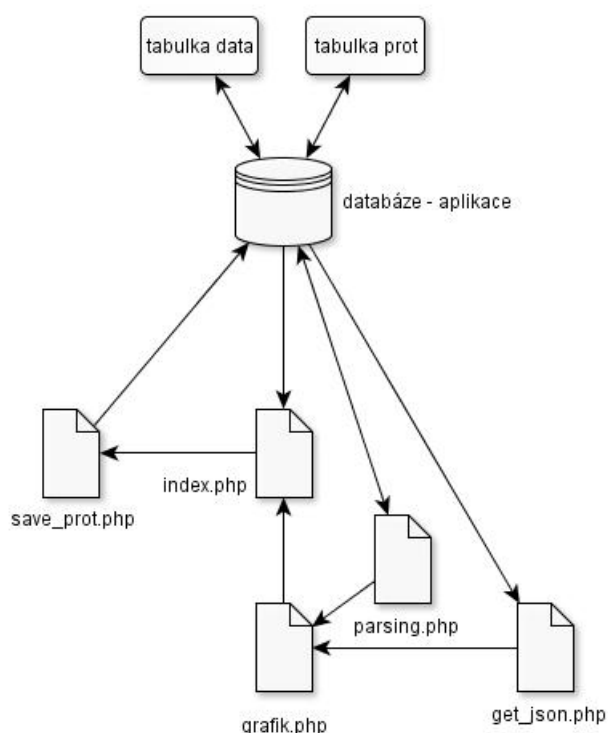
4 PRAKTICKÁ REALIZACE

4.1 Struktura aplikace

Aplikace v současnosti běží na virtuálním serveru WampServer. Jedná se o komplexní balíček, který obsahuje Apache 2.2, PHP 5.3 a MySQL 5.5 a utilitu PHPmyAdmin. Dovoluje tak testovat webové aplikace vyžadující PHP či MySQL na lokálním počítači. V případě, že vlastníme veřejnou IP (Internet Protocol) adresu je možné přistupovat k aplikaci odkudkoliv a zapojit do jejího testování více uživatelů. Nabízelo se použití alternativních systému jako VertrigoServ. Ten ovšem při svém běhu blokoval komunikační porty, na kterých běžely jiné programy. Proto byl upřednostněn WampServer, u kterého se tyto problémy nevyskytly.

Samotná aplikace se skládá z hlavní stránky `index.php`, která tvoří základní rozhraní pro komunikaci s uživatelem. K hlavní stránce jsou přidruženy podpůrné stránky, které se specializují již na dílčí úkony aplikace.

Grafické znázornění struktury aplikace viditelné z obr.4.1



Obr. 4.1: Struktura aplikace

4.1.1 Hlavní modul aplikace

Při tvorbě aplikace byla snaha vytvořit jednoduchý, přehledný a uživatelsky příjemný vzhled. V důsledku toho bylo zvoleno užití záložkového menu aplikace, které přispívá k celkové větší přehlednosti celku.

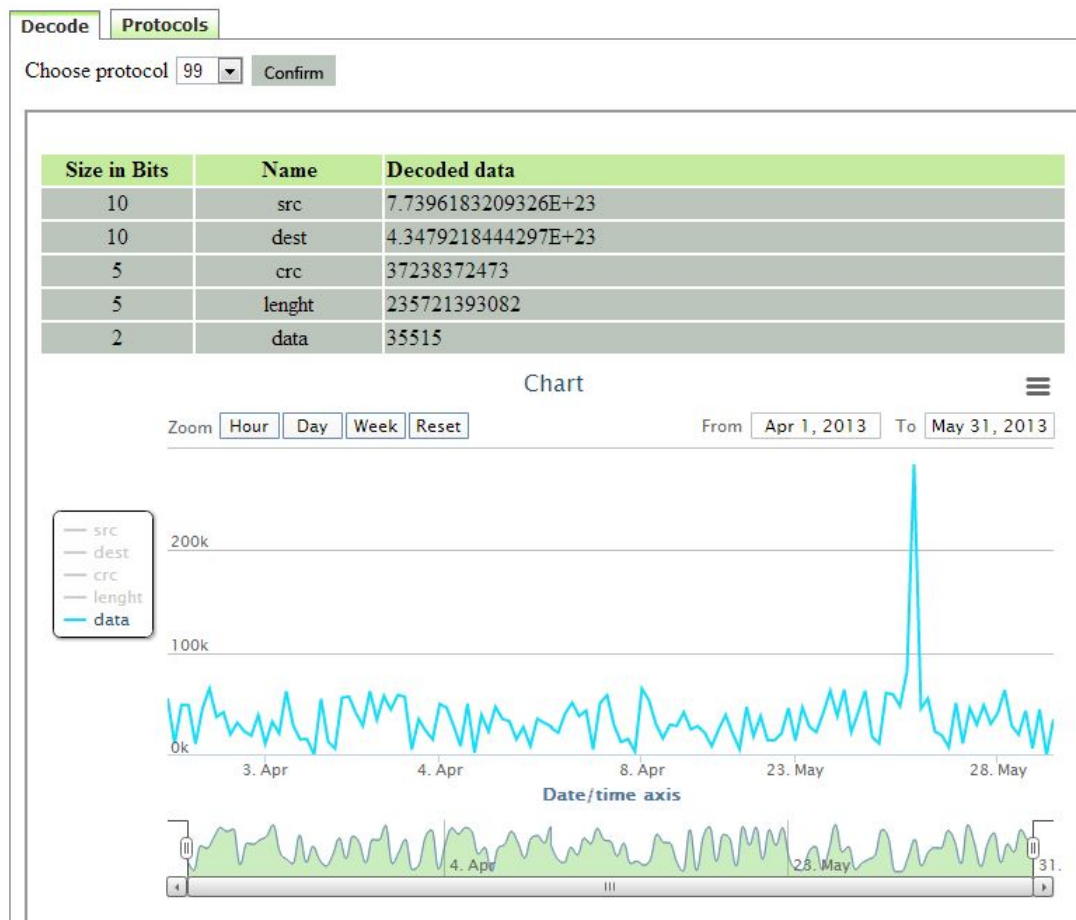
Menu aplikace se skládá ze dvou záložek. Záložka **Decode** uživateli umožňuje dekódovat data, zatímco v záložce **Protocols** je možné samostatné protokoly definovat. V případě rozšíření aplikace o další moduly lze tak snadno přidat záložky nové a celý systém obohatit o nové funkce bez narušení vzhledu její struktury.

K vytvoření grafického vzhledu aplikace bylo využito především kaskádových stylů CSS, které jsou deklarovány jednak uvnitř struktury HTML elementu, který ovlivňují. Případně pak v externím souboru pokud se jedná o vlastnosti globálnějšího charakteru, které ovlivňují více stránek najednou. Je výhodnější zapisovat do zvláštního souboru.

Deklarace je umístěna do dvou souborů:

- `tabcontent.css` – Zajišťuje základní vzhled a strukturu aplikace.
- `buttons.css` – Obsahuje grafické definice ovládacích prvků.

Adaptive Packets Decoder



Obr. 4.2: Hlavní okno aplikace

4.1.2 Modul definice protokolu

Modul definice protokolu se skrývá pod záložkou **Protocols**. V této části aplikace má možnost uživatel sám definovat vlastní strukturu protokolu, která bude užita při dekódování příchozích dat.

Celá struktura protokolu je zakotvena v jedinečném identifikačním čísle protokolu, které uživatel definuje v poli **Insert protocol code**. Zde uživatel definuje numerický kód protokolu, s kterým pak bude mít možnost dále pracovat a využít jej pro dekódování dat.

Po stisku tlačítka **Add input** může uživatel začít definovat protokol. Ten se skládá vždy ze dvojice údajů. Z jeho popisku v části **Variable** a jeho velikosti v bitech v poli **Size**. Maximální množství takto vytvářených dvojic je omezeno desíti. Jeli-kož je to dostatečná struktura pro vytvoření protokolu. Aby bylo možné dynamicky

přidávat počet polí, do kterých se struktura deklaruje. V závislosti na potřebách uživatele bylo potřeba vytvořit JavaScriptovou funkci, která dovoluje toto přidávání bez opětovného načtení celé aplikace. To přispívá k celkové uživatelské atraktivnosti, protože uživatel nemusí vyčkávat na to, než se celá aplikace aktualizuje a objeví se nová vstupní pole pro zadávání.

Ukázka této funkce:

```
<script language="Javascript" type="text/javascript">
var counter = 0;
var limit = 10; // max počet polí
function addInput(divName){
    if (counter == limit) { // kontrola počtu polí
        alert("You have reached the limit of adding" + counter
            + "inputs");
    }
    else { // jinak přidá dvě pole
        var newdiv = document.createElement('div');
        newdiv.innerHTML = "<br><input type='text' name='myInputs[]'
            size='8'>" +
            "<input type='text' name='myInputs2[]' size='1'>";
        document.getElementById(divName).appendChild(newdiv);
        counter++;
    }
}
</script>
```

Jakmile je uživatel spokojen se strukturou nadefinovaného protokolu může jej uložit pomocí stisku tlačítka **Save protocol** obr.4.3. Poté jsou data předána stránce **save_prot.php**, která zajišťuje uložení definovaného protokolu do databáze protokolů a uživatele informuje o zdárnosti tohoto procesu textovým výpisem obr.4.4. Po stisknutí tlačítka se dostaneme zpět do hlavního menu aplikace.

Adaptive Packets Decoder

Decode

Protocols

Define new protocol

Insert protocol code: 116

Variable	Size [Bits]
<input type="text" value="source"/>	<input type="text" value="20"/>
<input type="text" value="dest"/>	<input type="text" value="20"/>
<input type="text" value="seq"/>	<input type="text" value="5"/>
<input type="text" value="data"/>	<input type="text" value="4"/>
<input type="text" value="crc"/>	<input type="text" value="5"/>

Add Input

Save protocol

Obr. 4.3: Definice nového protokolu



Obr. 4.4: Oznámení o tom, že byl protokol uložen do databáze

4.1.3 Validace

Aby se předešlo omylům při zadávání protokolu, jako je například jeho uložení bez vyplnění nějakého ze vstupních polí, byla obsluha tlačítka **Save protocol** ošetřena JavaScriptovou funkcí. Tato funkce validuje, zda není vstupní pole před samotným uložení protokolu prázdné. V případě že ano, je uživatel informován o nutnosti prázdná pole vyplnit vyskakovacím oknem obr.4.5. To vše se děje ještě předtím, než je protokol uložen do databáze a to na straně klienta.

Adaptive Packets Decoder

Decode **Protocols**

Define new protocol

Insert protocol code:

Variable	Size [Bits]
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

Stránka na webu localhost říká:

#Error# You have to fill all empty fields

OK

Obr. 4.5: Chybové hlášení v případě nevyplnění vstupního pole

Funkce zajišťující validaci vstupních polí:

```
<script>
function validateForm()
{
var x=document.forms["formular"]["kod"].value;
var y=document.forms["formular"]["myInputs[]"].value;
var z=document.forms["formular"]["myInputs2[]"].value;
if (x==null || x=="")
{
alert("#Error# You have to fill all empty fields");
return false;
}
else if (y==null || y=="")
{
alert("#Error# You have to fill Variable fields");
return false;
}
else if (x==null || x=="")
{
alert("#Error# You have to fill Value fields");
return false;
}
}
</script>
```

4.1.4 Modul generování dat

Jedná se o součást aplikace zajišťující simulované generování dat a jejich ukládání do databáze. Simuluje tak skutečné přichozí pakety. Na základě použitého protokolu, je tak generovaný řetězec použit pro dekováání. Aktivace generátoru je svázáno s tlačítkem **Confirm**, které je možné nalézt v záložce **Decode** v hlavním menu aplikace. Po jeho stisknutí generátor vytvoří náhodný hexadecimální řetězec.

V generátoru dat je použita funkce `rand()`, která bez jakýchkoliv zadaných parametrů vrací náhodný řetězec v rozmezí 0 až 32768. Na tyto vygenerované řetězce je následně použito hashovacího algoritmu `sha1()`, který z vstupního řetězce vytváří

40 místný hexadecimální řetězec pevné délky.

V tomto generátoru je použito řetězců: `$val`, `$val1`, `$val2`, `$val3` a `$val4`. Na tyto řetězce je pak použit zmiňovaný algoritmus SHA-1. Těchto pět řetězců je nakonec spojeno do společného výstupního řetězce `$out` o velikosti 100 bajtů, který je následně ukládán do databáze. K opakovanému vytvoření nového náhodného řetězce dojde po opětovném stisknutí tlačítka.

Zdrojový kód generátoru dat:

```
<html>
<head>
</head>
<body>
<?php
include( 'config.php' ); // konfigurace spojení s databází
$val = ''; // inicializace pomocných proměnných
$val1='';
$val2='';
$val3='';
$val4='';
for ( $i=0;$i<1;$i++ ){
    $val.= rand(); // generování náhodných dat
    $val1.= rand();
    $val2.= rand();
    $val3.= rand();
    $val4.= rand();

    $value=sha1($val); // hashovací funkce sha1-> 40hex
    $value1=sha1($val1);
    $value2=sha1($val2);
    $value3=sha1($val3);
    $value4=sha1($val4);
    $out=($value.$value1.$value2.$value3.$value4);
    echo($out); // výstup na obrazovku
    $insert=("INSERT INTO data (data) VALUES ( '$out' )");
    mysql_query($insert) // uložení dat do DB ?>
```

4.1.5 Modul dekódování dat

Tato část aplikace se stará o samotné dekódování příchozích dat, které se vytvářejí v **modulu generování dat**. O tuto záležitost se stará PHP skript umístěný v souboru `parsing.php`.

Z databáze, kde jsou simulovaná data ukládány, se vybere vždy řetězec s nejvyšší hodnotou proměnné `id_data`, která tak zastupuje vždy nejnovější vygenerovaný řetězec. Tímto je zajištěno to, že se z databáze příchozích dat vybere vždy nejnovější záznam. Zároveň se z databáze, kam jsou ukládány struktury protokolů vybere ten, který si uživatel zvolil z rozevíratelné nabídky a řetězec je dekódován podle této struktury. Výsledek operace dekódování je poté vrácen ve formě přehledné tabulky obr.4.6 do hlavního okna aplikace, záložka **Decode**, která mimo dekódování data obsahuje také strukturu protokolu (názvy jednotlivých polí včetně jejich velikostí), pomocí níž byla data dekódována.

Aby mohl být výsledek dekódovací operace vrácen do daného umístění v aplikaci

Size in Bits	Name	Decoded data
4	source	3220459535
4	dest	1223695047
4	nwkAdd	2354883500
8	macAdd	9.4301430301511E+18
1	seq	164
30	data	1.4409875271352E+72
2	crc	37481

Obr. 4.6: Dekódovaná data

bez nutnosti znovu načíst celé prostředí aplikace, které by bylo pro uživatele nepohodlné, bylo užito vnořeného plovoucího rámu `<iframe>`, který ve svém parametru obsahuje zdroj, odkud se má jeho obsah načítat. V tomto případě se jedná o výstup z modulu pro dekódování dat.

Samotné dekódování a práce s modulem probíhá v záložce **Decode**. Uživatel zde má možnost vybrat z rozevíratelného menu **Choose protocol** číslo protokolu, který si sám zadefinoval, případně si vybrat z nabídky protokol jiný, který zde byl už přednastaven. Protokoly jsou řazeny vzestupně od nejnižšího k nejvyššímu. Načítání protokolů do rozevíratelné nabídky je řešeno dynamicky z databáze. Uživatel se tak nemusí obávat toho, že při ukončení aplikace by jím definované protokoly zmizely a on je při příštím spuštění musel deklarovat znovu. Po stisku tlačítka **Confirm** se

zobrazí tabulková struktura, která zobrazuje jednak strukturu protokolu zvoleného z rozevratelné nabídky, ale také poslední dekodovaný řetězec daného protokolu. V případě, že chce uživatel zvolit jiný protokol, vybere znovu z rozevratelného menu požadovaný protokol a opět potvrdí výběr tlačítkem **Confirm**.

4.1.6 Grafický modul

Aby mohla být dekodovaná data ještě přehledněji uživatelům zobrazována, byl zde implementován grafický modul HighStock, který vizualizuje v podobě grafu příchozí data.

Pro tvorbu grafů, které je možno zobrazovat na webových stránkách dnes existuje nespočet knihoven, které tuto činnost nějakým způsobem řeší. Mezi nejznámější knihovny pro tvorbu grafů patří Google Chart Api, Plotkit, HighCharts, Flot a mnoho dalších. Výsledkem je pak většinou vygenerovaný obrázek ve formátu JPEG (Joint Photographic Experts Group), PNG (Portable Network Graphics). Zvláštní skupinou jsou potom grafy vytvářené pomocí technologie Flash.

Zpočátku bylo při tvorbě grafického modulu využito rozhraní Google Chart, které nabízí široké možnosti tvorby grafů. Práce s ním je velmi jednoduchá. Google Chart Api má tu výhodu, že graf se generuje pomocí speciálně upravené URL (Uniform Resource Locator), ve které se posílají jako argumenty všechna nastavení výsledného grafu.

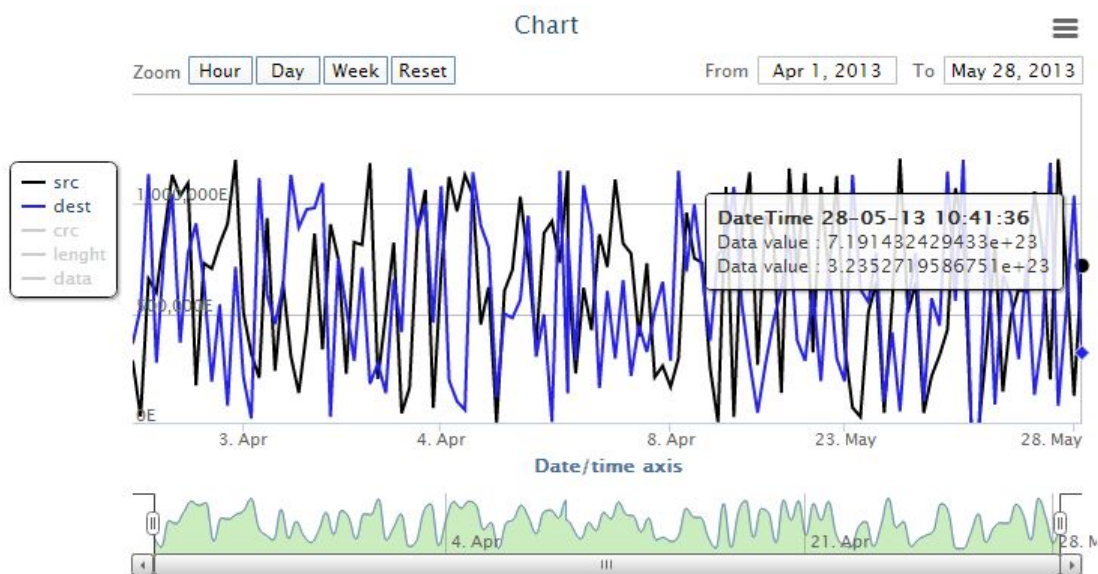
Vložení grafu je realizováno v HTML kódu prostřednictvím tagu pro vkládání obrázků ``, kde se do argumentu *src* zapíše jeho URL. Google pak graf vygeneruje a zašle jej na požádání zpět do stránky. Z toho pramení i jedna nevýhoda tohoto rozhraní a to nemožnost generovat grafy bez připojení k internetu. Google Chart Api je však skvělým a dostačujícím nástrojem pro tvorbu jednoduchých grafů na webu.

Bohužel funkce omezené na úkor jednoduchosti tohoto rozhraní se projeví jako limitují při pozdějším vývoji grafického modulu. Jako jeho náhrada byla zvolena knihovna HighCharts, konkrétně pak její část HighStock, která dovolila větší míru customizace.

Použitá HighStock knihovna

Celý grafický modul je součástí hlavního okna aplikace. Uživatel si vybere z rozevratelné nabídky číslo protokolu, pro který chce zobrazit dekodovaná vizualizovaná data a toto potvrdí stiskem tlačítka **Confirm**. Následně se mu do grafu vykreslí data požadovaného protokolu.

Data, která jsou potřebná k vykreslení grafu, jsou načítána ze souboru `get_json.php`, který se stará o načítání dat z databáze a upravuje ho do JSON (JavaScript Object Notation) formátu, s kterým dokáže pracovat knihovna HighStock. Samotný graf je vložen do vnořeného plovoucího rámu `<iframe>`, jehož obsah je načítán ze souboru `grafik.php` (obr.4.7).



Obr. 4.7: Grafický modul

Funkce grafického modulu

- **Zoom** - Kliknutím a tažením myši nad určitou oblastí průběhu grafu dojde k jeho přiblížení. Pro opuštění zoom funkce slouží tlačítko reset v levém horním rohu.
- **Navigátor** - Ve spodní části časové osy (Date/Time axis) jsou umístěny dva posuvníky, kterými si lze separovat určitou část průběhu.
- **Range Selector** - Slouží k rychlému časovému vyfiltrování. Je možno zvolit ze tří přednastavených hodnot a to: Hour (poslední hodina), Day (posledních 24 hodin) a Week (poslední týden). Tlačítkem Reset se vrátí průběh grafu do původního pohledu.
- **Legenda** - Zobrazuje nabídku průběhů možných k zobrazení. Defaultně jsou všechny průběhy vypnuty a uživatel si může zvolit právě ty, o které má zájem. Je možné mít zapnuto více průběhů najednou. Nachází se v levé části vedle ypsilonové osy.

- **Selector** - Umožňuje nastavit datové rozmezí, pro které se má průběh vykreslit.
- **Export Button** - Pod tímto tlačítkem se skrývá možnost vytisknout aktuálně zobrazovaný graf nebo jej uložit do formátu JPEG, PNG, PDF (Portable Document Format) či vektorového SVG (Scalable Vector Graphics).
- **Label Marker** - Po najetí kurzoru nad jednotlivé body grafu se zobrazí aktuální hodnota a čas.

4.1.7 Databáze

K funkčnosti celé aplikace bylo zapotřebí nějakým způsobem uchovávat data. Ať už se jedná o simulovaná data generátoru, dekodovaná data nebo údaje o tom, jak vypadají struktury protokolů, které uživatelé vytváří. Velkou výhodou tvorby aplikace v skriptovacím jazyku PHP byl fakt, že je úzce spjat s relační databází MySQL, která tvoří hlavní jádro celé aplikace.

Pro jednoduchou správu databáze byl použit programový systém phpMyAdmin. Jde o nástroj, který je napsaný v jazyce PHP, který dovoluje jednoduše spravovat obsah databází prostřednictvím uživatelsky příjemného webového rozhraní. Umožňuje databáze vytvářet, rušit, vyprazdňovat či přímo editovat jejich strukturu změnou datových typů, přidáváním záznamů či sloupců. Navíc je tento správce databází včleněn do balíku Wampserver, který byl při vývoji aplikace použit.

Umění jak správně databázi navrhnout, aby byla co nejefektivnější a nebyly v ní zbytečné redundance, není lehký úkol a vyžaduje to často velké zkušenosti.

Databáze, která byla použita v této bakalářské práci se skládá ze dvou tabulek. První tabulka nese označení **data** (obr.4.8). Jejím úkolem je uchovávat informace, které jsou vytvářeny generátorem simulovaných dat. Dále jsou pak v ní uchovávány data, která byla získána při dekodování dat. Druhá tabulka s názvem **prot** (obr.4.9) uchovává informace o tom, jakou strukturu mají protokoly, které uživatelé definují. Je nesmírně důležité, aby tyto záznamy byly nějakým způsobem uchovávány a byly tak dostupné i v případě, kdy se aplikace ukončí a znovu zapne.

Připojení k databázi ze souborů s příponou PHP je realizováno pomocí souboru `config.php`, který je v každé části aplikace (každé podstránce) vložen pomocí příkazu `include()`. Toto je velmi výhodné, jelikož do každého souboru není třeba vepisovat řádky kódu zajišťující tutéž činnost. Tento konfigurační soubor obsahuje umístění databáze, jméno s heslem uživatele a název databáze.

Konfigurační soubor pro připojení databáze

```
<?php
/* Zobrazování chybových hlášek */
error_reporting(0);

/* DATABÁZE */
$db_server = "localhost"; // adresa, kde běží MySQL server
$db_user   = "root";       // uživatel
$db_pass   = "123456";     // heslo
$db_name   = "aplikace";   // jméno databáze

$connect = mysql_connect($db_server, $db_user, $db_pass);
// spojení se serverem MySQL
mysql_select_db($db_name, $connect); // spojení s databází

//nastavení znakové sady
mysql_query("SET character_set_client=utf8");
mysql_query("SET character_set_connection=utf8");
mysql_query("SET character_set_results=utf8");
?>
```

Struktura tabulky *data*:

- **id_data** (int) – identifikátor záznamů
- **paket** (text) – řetězec generované modulem generování dat
- **cas** (timestamp) – čas uložení záznamu.

Ostatní sloupce tabulky se dynamicky vytvářejí podle zvoleného protokolu.

Struktura tabulky *prot*:

- **id_p** (int) – identifikátor záznamu
- **kod** (int) – jedinečný kód protokolu
- **popis** (text) – hodnota Variable v definici protokolu
- **size** (int) – hodnota Size v definici protokolu

id_data	paket	cas	cp	src	dest	crc	data	lenght
524	994a4ebbf31d6a54aea312295bc1f265cb8c1401f9f6522a5c...	2013-05-23 16:39:15	99	4a4ebbf31d6a54aea312	295bc1f265cb8c1401f9	f6522a5c82	3845	ae0609d7a7
13	993d30545b87b50f25c1b14d3b62609754ec6afd264df7adde...	2013-04-01 21:45:34	99	3d30545b87b50f25c1b1	4d3b62609754ec6afd26	4df7adde87	d9e3	b80f10c7a5
14	990757b64adfb4b4ddac7d700750a2f2e4020489feeffcd19f...	2013-04-01 21:45:36	99	0757b64adfb4b4ddac7d	700750a2f2e4020489fe	effcd19f7c	2e4b	474705ebfe
15	998b8c9c598a2d41e09b3bf05bd1075dd5123308819f30f35...	2013-04-01 21:45:39	99	8b8c9c598a2d41e09b3b	f05bd1075dd512330881	9f30f359d	c12c	6d30ee8fa4
527	990f270789f51c1c7c39fefab6a5ef1ecf1be367ec5da211b...	2013-05-23 16:45:48	99	0f270789f51c1c7c39f	efab6a5ef1ecf1be367e	c5da211bb0	b7e4	51cecf58b9
20	997e901182ce8ce7c8aaad3aef1437519b64633425fe773fe...	2013-04-01 21:45:49	99	7e901182ce8ce7c8aaad	3aef1437519b6463342	5fe773fe0b	c05b	1590aa8680
21	99b7e1ba37a7894fa73d86a840b42373ee3b4ebfa6c50b8487...	2013-04-01 21:45:53	99	b7e1ba37a7894fa73d86	a840b42373ee3b4ebfa6	c50b8487b5	2d88	3261c131b0
23	99ef82459296e752e6c816dd2f5641fc5183ebe31caa56c3f2...	2013-04-01 21:45:55	99	ef82459296e752e6c816	dd2f5641fc5183ebe31c	aa56c3f208	b2ce	97f7199d81
539	990716d9708d321ffb6a00818614779e779925365cc1901f3e...	2013-05-23 17:13:29	99	0716d9708d321ffb6a00	818614779e779925365c	c1901f3e94	6e2c	24fc782d1b
28	99db9804f6a9e1f1794ba44e3aee7f356ff663797167f45d33...	2013-04-01 23:32:57	99	db9804f6a9e1f1794ba4	4e3aee7f356ff6637971	67f45d331d	f09	f34eb29e10
541	996b64fc4a505db0514e81756f20aa6e3c22e7fa627a690212...	2013-05-23 17:13:44	99	6b64fc4a505db0514e81	756f20aa6e3c22e7fa62	7a6902129c	596a	407fc44d32
542	99884556be30cc66846fd0a4cd41d0488d1811e6c74e643d4b...	2013-05-23 17:13:47	99	884556be30cc66846fd0	a4cd41d0488d1811e6c7	4e643d4bc4	a3a5	ca6792dacd
287	99b50f4bb1228de5a38b8b62e4a186690b50472c27e6518fa4...	2013-04-09 12:21:11	99	b50f4bb1228de5a38b8b	62e4a186690b50472c27	e6518fa4d0	543f	b5d33c3320
543	990ac5209abecab302643f15fd9aba410c4b0f8f8ef8c592d5...	2013-05-23 17:13:52	99	0ac5209abecab302643f	15fd9aba410c4b0f8f8e	f8c592d57c	f65f	a16597b501
544	99735e0b341566ac7b7efa58429ece31812ac6a3da2b242bc6...	2013-05-23 17:13:58	99	735e0b341566ac7b7efa	58429ece31812ac6a3da	2b242bc685	959e	7a517fc0f9
547	99f28670fd583d3f27db30ca0d405e8c14d6baab34c4c98d4...	2013-05-23 17:33:19	99	f28670fd583d3f27db3	0ca0d405e8c14d6baab3	4c4c98d454	fb64	56f28e3f32
548	996b64fc4a505db0514e81756f20aa6e3c22e7fa627a690212...	2013-05-23 17:33:51	99	6b64fc4a505db0514e81	756f20aa6e3c22e7fa62	7a6902129c	596a	407fc44d32

Obr. 4.8: Obsah tabulky data

id_p	kod	popis	size
149	0	source	4
150	0	dest	4
151	0	nwkAdd	4
152	0	macAdd	8
153	0	seq	1
154	0	data	30
155	0	crc	2
210	99	src	10
211	99	dest	10
212	99	crc	5
213	99	lenght	5
214	99	data	2
139	100	src	15
140	100	dst	15
141	100	length	10
142	100	seq	10
143	100	crc	10
144	100	data	40

Struktura protokolu číslo 0

Struktura protokolu číslo 99

Struktura protokolu číslo 100

Obr. 4.9: Obsah tabulky prot

5 ZÁVĚR

Při tvorbě projektu byl kladen důraz na využití moderních technologií pro vývoj a tvorbu webových aplikací. V první části práce byly charakterizovány technologie, které byly při tvorbě aplikace použity.

Byl vytvořen uživatelsky příjemný vzhled aplikace s možností jejího dalšího rozšiřování bez zásadního zasahování do její struktury. To především díky záložkovému systému celé aplikace, u kterého lze přidávat nové záložky a pod nimi pak realizovat další funkce. Tímto je zajištěna určitá modularita.

Aplikace je schopná náhodně generovat simulované řetězce dat, které zastupují pakety a ty ukládat do databáze. Následně je možnost s nimi dále pracovat (dekódovat, vykreslovat do grafu).

Modul definice protokolů dovoluje uživateli vytvářet libovolné struktury protokolů, které je možné v aplikaci nalézt pod číselným kódem protokolu. Tyto protokoly jsou pak společně s generovanými daty ukládány do databáze, a tak uživatel po opětovném spuštění aplikace o svá data nepřijde.

Modul dekodování přehledně zobrazuje data, která jsou zpracována podle zvoleného protokolu v hlavním menu aplikace.

Grafický modul zobrazuje příchozí dekodovaná data v grafické závislosti v závislosti na čase. Průběh grafu je možno filtrovat, vybírat průběhy k zobrazení.

LITERATURA

- [1] CASTAGNETTO, J. *Programujeme PHP profesionálně*. 2. oprav. a aktualiz. vyd. Brno: Computer Press, 2004, 656 s. ISBN 80-722-6310-2.
- [2] HAUZAR, D. *Tvorba databází v MySQL - I* [online]. 20.3.2003 [cit 26-11-2012]. Dostupný z WWW: <<http://www.abclinuxu.cz/clanky/navody/tvorba-databazi-v-mysql-i>>
- [3] HREBENAR, J. *Úvod do JavaScriptu* [online]. 24.3.2010 [cit 24-11-2012]. Dostupný z WWW: <<http://www.pestujemeweb.cz/obsah/javascript/javascript-uvod.php>>
- [4] JANOVSÝ, D. *Možnosti PHP* [online]. 12.2.2009 [cit 14-10-2012]. Dostupný z WWW: <<http://www.jakpsatweb.cz/php/moznosti-php.html>>
- [5] KOSEK, J. *Příkaz foreach* [online]. 12.2.2009 [cit 17-11-2012]. Dostupný z WWW: <<http://www.kosek.cz/clanky/php4/ch01s05.html>>
- [6] PÍSEK, S. *JavaScript: efektní nástroj oživení www stránek*. 1. vyd. Praha: Grada, 2001, 231 s. ISBN 80-247-0014-X.
- [7] PONKRÁC, M. *PHP a MySQL: bez předchozích znalostí*. Vyd. 1. Brno: Computer Press, 2007, 221 s. ISBN 978-80-251-1758-3.
- [8] STANÍČEK, P. *CSS Kaskádové styly: kompletní průvodce*. 2. vyd. Brno: Computer Press, 2003, 178 s. ISBN 80-722-6872-4.
- [9] ŠIMEK, Milan. skripta: *Bezdrátové senzorové sítě*. Brno, 2013, 165 s. ISBN 978-80-214-4638-0.
- [10] ZAJÍC, P. *PHP a vkládání záznamů do databází* [online]. 20.8.2004 [cit 24-11-2012]. Dostupný z WWW: <http://www.linuxsoft.cz/article.php?id_article=357>
- [11] ZRALÝ, J. *jQuery pro začátečníky: základy* [online]. 25.9.2008 [cit. 4-12-2012]. Dostupný z WWW: <<http://citron.blueboard.cz/clanek/jquery-pro-zacatecniky-zaklady>>
- [12] *ABClinuxu.cz* [online]. 16.11.2012 [cit. 30-11-2012]. refresh pomocí selectu. Dostupné z WWW: <<http://www.abclinuxu.cz/poradna/programovani/show/369121>>

- [13] *dynamicsdrive.com* How to populate php/html form with MySQL data. [online]. 6.12.2009 [cit. 2013-05-4]. Dostupné z WWW: <<http://www.dynamicdrive.com/forums/showthread.php?45895-How-to-populate-php-html-form-with-MySQL-data>>
- [14] *highcharts.com* Highstock Options Reference [online]. 2012 [cit. 2013-04-21]. Dostupné z WWW: <<http://api.highcharts.com/highstock>>
- [15] *highcharts.com* Highcharts and Highstock documentation [online]. 2012 [cit. 2013-04-21]. Dostupné z WWW: <<http://docs.highcharts.com/#home>>
- [16] *jakpsatweb.cz* Přehled vlastností CSS. [online]. 6.12.2012 [cit. 2013-05-2]. Dostupné z WWW: <<http://www.jakpsatweb.cz/css/css-vlastnosti-hodnoty-prehled.html>>
- [17] *tutorialy.vytvorweb.cz* Formuláře - Javascript [online]. 2008 [cit. 2013-03-17]. Dostupné z WWW: <<http://tutorialy.vytvorweb.cz/javascript/formulare.html?text1=&text2=>>
- [18] *webdesignledger.com* 13 Useful JavaScript Solutions for Charts and Graphs [online]. 9.2.2010 [cit. 2013-05-2]. Dostupné z WWW: <<http://webdesignledger.com/resources/13-useful-javascript-solutions-for-charts-and-graphs>>
- [19] *zaachi.com* Dynamické přidávání elementů formuláře. [online]. 15.6.2008 [cit. 2013-05-6]. Dostupné z WWW: <<http://www.zaachi.com/cs/items/dynamicke-pridavani-elementu-formulare.html#>>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

CRC	Cyclic Redundancy Check
CSS	Cascading Style Sheets
DOM	Document Object Model
FCS	Frame Control Sequence
GPL	General Public License
HTML	HyperText Markup Language
IP	Internet Protocol
JPEG	Joint Photographic Experts Group
JSON	JavaScript Object Notation
MySQL	My Structured Query Language
PDF	Portable Document Format
PHP	Hypertext Preprocessor
PNG	Portable Network Graphics
SHA-1	Secure Hash Algorithm 1
SQL	Structured Query Language
SVG	Scalable Vector Graphics
URL	Uniform Resource Locator
XHTML	Extensible HyperText Markup Language
XML	Extensible Markup Language

SEZNAM PŘÍLOH

A médium CD

36

A MÉDIUM CD

Na přiloženém CD nosiči je umístěna elektronická verze bakalářské práce a v adrese *www* pak samostatné zdrojové soubory webové aplikace. Tyto soubory je potřeba nakopírovat na webhostingový server s podporou PHP a MySQL. Pro správnou funkci aplikace je nezbytné nakonfigurovat databázi. K tomu slouží soubor *aplikace.sql*, který stačí importovat do MySQL databáze.