

**VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky**

**Předplacené hlasové služby na platformě openBTS
Pre-paid Voice Services Based on openBTS Platform**

2016

Bc. Ladislav Beháň

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Zadání diplomové práce

Student: **Bc. Ladislav Beháň**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2601T013 Telekomunikační technika

Téma: **Předplacené hlasové služby na platformě openBTS**
Pre-paid Voice Services Based on openBTS Platform

Jazyk vypracování: čeština

Zásady pro vypracování:

1. Úvod do projektu openBTS.
2. Open-source účtovací systémy.
3. Návrh a realizace účtování služeb.
4. Řízení přístupu ke službám na základě účtování.
5. Závěr a zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

- [1] Madsen, L., Bryant, R., Meggelen, J., "Asterisk: The Definitive Guide." Publisher: O'Reilly Media, Fourth Edition edition, 846 pages, 2013. ISBN-13: 978-1449332426
- [2] M. Vozňák, "Technologie a protokoly multimediálních komunikací pro integrovanou výuku VUT a VŠB-TUO." VŠB-TU Ostrava, 2014, 252 str., ISBN 978-80-248-3326-2.


Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí diplomové práce: **doc. Ing. Miroslav Vozňák, Ph.D.**

Datum zadání: 01.09.2015

Datum odevzdání: 29.04.2016




doc. Ing. Miroslav Vozňák, Ph.D.
vedoucí katedry


prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: 13. března 2016


.....
podpis studenta

Poděkování

Rád by som poďakoval doc. Ing. Miroslavovi Vozňákovi, Ph.D. za odbornú pomoc, prejavenu ochotu a konzultácie pri vytvaraní tejto diplomovej práce.

Abstrakt

Táto diplomová práca nadväzuje na moju bakalársku prácu s názvom OpenBTS a její integrace do telekomunikační sítě. V práci sa detailne venujem konfigurácii tohto projektu a možnosti prepojenia medzi viacerými Asterisk systémami. Využitím rôznych nástrojov je v diplomovej práci implementovaný vlastný účtovací systém, na základe ktorého môžem z OpenBTS vytvoriť nízkonákladového mobilného operátora.

Klíčová slova

GSM, OpenBTS, Asterisk, Smqueue, Subscriber Registry, SIP, IAX, USRP , C++, PHP, HTML, CSS, trunk, VoIP

Abstract

This diploma thesis knots on my bachelor thesis titled OpenBTS and its Integration into Telecommunication Network. This work gives detailed explanation of the project configuration and describes connectivity options with multiple Asterisk systems. As the title implies, this diploma thesis presents implementation of its own billing system, which in collaboration with OpenBTS system can be operated as a low-cost mobile operator.

Keywords

GSM, OpenBTS, Asterisk, Smqueue, Subscriber Registry, SIP, IAX, USRP , C++, PHP, HTML, CSS, trunk, VoIP

Zoznam použitých skratiek

Skratka	Anglický názov	Slovenský názov
ADC	Administrative Centre	Administratívne centrum
AEL	Asterisk Extension Language	Programovací jazyk Asterisku
AGI	Asterisk Gateway Interface	Rozhranie pre prevádzku dialplánu
AMI	Asterisk Managment Interface	Rozhranie pre správu Asterisku
BSC	Base Station Controller	Základňová riadiaca jednotka
BSS	Base Station Sub-System	Subsystem základňových staníc
BTS	Base Tranceiver Station	Základňová stanica
CDR	Calling Data Records	Záznamy hovoru
CLI	Command Line Interface	Príkazový riadok
DCA	Dynamic Channel Allocation	Dynamické pridelovanie kanálov
DID	Direct Inward Dialing	Priama predbvoľba
FCA	Fixed Channel Allocation	Fixné pridelovanie kanálov
FEC	Forward Error Correction	Dopredná oprava chýb
FPGA	Field-programmable gate array	Programovateľné hriadľové pole
FTP	File Transfer Protocol	Protokol pre prenos súborov
GIF	Graphics Interchange Format	Grafický formát rastrovej grafiky
GSM	Global System for Mobile Communication	Globálny systém pre mobilnú komunikáciu
HLR	Home Location Register	Databáza domovských účastníkov
HTTP	Hypertext Transfer Protocol	Hypertextový prenosový protokol
IMAP	Internet Message Access Protocol	Internetový protokol umožňujúci prístup k e-mailovým schránkam
IMSI	International Mobile Subscriber Identity	Medzinárodné identifikačné číslo
ISDN	Integrated Services Digital Network	Digitálna sieť integrovaných služieb
IVR	Interactive Voice Response	Interaktívna hlasová odozva
LAI	Local Area Identification	Kód lokalizačnej oblasti
MS	Mobile Station	Mobilná stanica
MSC	Mobile Switching Center	Mobilná rádiatelefónna ústredňa
NMC	Network Monitoring Center	Centrum managementu siete
NSS	Network Switching Subsystem	Sieťový spojovací subsystem
OSS	Operational and Support Subsystem	Operačný a podporný subsystem
PIN	Personal Identification Number	Osobné identifikačné číslo

PLMN	Public Land Mobile Network	Verejná pozemná mobilná sieť
PNG	Portable Network Graphics	Prenosná sieťová grafika
PSTN	Public Switched Telephone Network	Verejná telefónna sieť
SCCP	Signalling Connection Control Part	Protokol siet'ovej vrstvy v signalizačnom systéme SS7
SDR	Software Defined Radio	Softwarovo definované rádio
SIP	Session Initiation Protocol	Protokol pre inicializáciu relácií
SMSC	Short Message Service Center	Stredisko krátkych textových správ
SMTP	Simple Mail Transfer Protocol	Jednoduchý protokol na prenos pošty
TCH	Traffic Channels	Prevádzkové kanály
TDM	Time Division Multiplexing	Viacnásobný prístup s časovým delením
TMSI	Temporary Mobile Subscriber Identity	Dočasná identifikácia mobilného účastníka
TRX	Transceiver	Vysielač
USRP	Universal Software Radio Peripheral	Univerzálny softwarovo riadený
UHD	USRP Hardware Driver	USRP hardwarový ovládač
USB	Universal Serial Bus	Univerzálna sériová zbernica
VoIP	Voice over Internet Protocol	Internetová telefónia
XHTML	Extensible Hypertext Markup Language	rozšíriteľný hypertextový značkový jazyk
XML	eXtensible Markup Language	rozšíriteľný značkovací jazyk

Obsah

1	Úvod.....	1
2	Úvod do projektu OpenBTS.....	2
2.1	Architektúra projektu OpenBTS.....	2
2.1.1	USRP (Universal Software Radio Peripheral).....	3
2.1.2	Vrstvový model OpenBTS	3
2.1.3	Hardwarové požiadavky	4
2.1.4	Smqueue	4
2.1.5	Subscriber registry.....	5
2.1.6	OpenBTS Command Line Interface (CLI).....	5
2.1.7	Databáze používané v systéme OpenBTS	5
2.1.8	Asterisk.....	6
2.1.9	Asterisk Managment Interface.....	6
2.1.10	Asterisk Gateway Interface	7
2.1.11	Protokol SIP.....	7
2.1.12	SIP komunikácia.....	8
2.1.13	Protokol IAX	9
2.1.14	IAX komunikácia	9
3	Open-source účtovacie systémy.....	10
3.1	A2Billing.....	10
3.1.1	Inštalácia účtovacieho systému A2Billing.....	11
3.1.2	Popis základných funkcií systému A2Billing.....	14
3.2	Freeside	16
3.3	EasyITSP	17
3.4	astCDRview.....	17
3.5	AstPP	17
3.6	CitrusDB.....	17
3.7	ASTERISELL	18

4	Návrh a realizácia účtovania služieb	20
4.1	Inštalácia OpenBTS a jeho súčastí	20
4.2	Inštalácia a konfigurácia Asterisk PBX.....	23
4.3	Ďalšie použité technológie	26
4.3.1	Apache.....	26
4.3.2	PHP.....	26
4.3.3	MySQL.....	27
4.3.4	phpMyAdmin	27
4.3.5	PHP JpGraph	27
4.3.6	FPDF.....	28
4.4	Automatická registrácia IMSI čísiel v sieti OpenBTS	28
4.5	Spôsob účtovania za hlasové služby v sieti OpenBTS.....	32
4.6	Kontrola prekročenia predplatených služieb pri hovore.....	35
4.7	Spôsob kontroly odoslaných textových správ	38
5	Riadenie prístupu k službám na základe účtovania	40
6	Záver a zhodnotenie dosiahnutých výsledkov	44
7	Použitá literatúra.....	46
8	Zoznam príloh.....	48

Zoznam obrázkov

Obrázok 1: Architektúra systému OpenBTS	3
Obrázok 2: USRP N210.....	4
Obrázok 3: Databáze a ich komunikácia s komponentami OpenBTS	6
Obrázok 4: Príklad zachytenej SIP komunikácie protokolovým analyzátorom Wireshark.....	8
Obrázok 5: Príklad zachytenej komunikácie medzi dvoma Asterisk servermi (IAX trunk)	9
Obrázok 6: Schéma funkčnosti A2Billing systému.....	10
Obrázok 7: Prihlasovacia stránka systému A2Billing	14
Obrázok 8: Základné funkcie systému A2Billing.....	14
Obrázok 9: Diagram funkčnosti softwaru CitrusDB	18
Obrázok 10: Schéma zapojenia.....	29
Obrázok 11: Manuálna registrácia IMSI.....	29
Obrázok 12: Automatická registrácia v sieti openBTS.....	30
Obrázok 13: SMS správa novému zákazníkovi	31
Obrázok 14: Priebeh programu cdr_manager.cpp	35
Obrázok 15: Notifikačná textová správa o nízkom stave kreditu	35
Obrázok 16: Priebeh programu call_manager.cpp.....	36
Obrázok 17: Notifikačná sms o nulovom stave kreditu.....	37
Obrázok 18: Priebeh programu sms_manager.cpp	39
Obrázok 19: Logo operátora	40
Obrázok 20: Prihlasovacia stránka OpenBTS.....	40
Obrázok 21: Uvítací odkaz na webových stránkach operátora.....	41
Obrázok 22: Hlavné menu	41
Obrázok 23: Aktivácia predplatených hlasových služieb	41
Obrázok 24: Platba vybranej služby prostredníctvom PayPal	42
Obrázok 25: Prehľad všetkých uskutočnených hovorov	42
Obrázok 26: Počet hovorov v jednotlivých mesiacoch.....	43

Obrázok 27: Počet odoslaných textových správ v jednotlivých mesiacoch	43
Obrázok 28: Celkové množstvo investovaných peňazí do hlasových služieb.....	43

Zoznam tabuliek

Tabuľka 1:	Štruktúra MySQL databáze users.db	31
Tabuľka 2:	Príklady niektorých CDR záznamov	33
Tabuľka 3:	Ceny hovoru pri jednotlivých scenároch použitia.....	34
Tabuľka 4:	Štruktúra MySQL databáze USER_data.db.....	34

1 Úvod

Realizácia vlastnej GSM siete prostredníctvom open-source unixovej aplikácie? V minulosti možno neuskutočniteľná predstava, avšak v súčasnosti sa open-source komunita zaoberá omnoho viac touto problematikou. Výhodou takejto realizácie je jej lacnejšia prevádzka, ľahšie by sa pristupovalo a prispôbovalo jednotlivým potrebám užívateľa a dohľad nad kompletným systémom by bol značne jednoduchší.

Vo svojej bakalárskej práci som venoval OpenBTS a jej integrácii do telekomunikačnej siete, kde som overal funkčnosť systému prostredníctvom zasielania textových správ a uskutočňovania telefónnych hovorov v rámci OpenBTS siete alebo medzi mobilným telefónom a softwarovým telefónom nachádzajúcim sa na druhej strane SIP/IAX trunku. Prostredníctvom OpenBTS je možné si vytvoriť svoju vlastnú GSM sieť, ktorej náklady nepresahujú desatinu prevádzkových nákladov súčasnej GSM siete a navyše s možnosťou pripojenia k internetu by si užívatelia mohli v rámci tejto siete telefonovať po celom svete. Pri jej písaní ma napadla myšlienka, či by som si nemohol pomocou tohto systému v spolupráci s ďalšími softwarovými nástrojmi vytvoriť svojho mobilného operátora prostredníctvom zavedenia predplatených služieb. A práve táto myšlienka sa stala cieľom mojej diplomovej práce.

Diplomová práca je rozdelená do piatich častí. V prvej kapitole sa budem venovať úvodu do projektu OpenBTS, predstavím myšlienku tohto projektu, princíp fungovania, z akých prvkov sa architektúra systému skladá a popíšem jednotlivé požiadavky pre realizáciu vlastnej GSM siete práve prostredníctvom tohto systému. V druhej časti sa zameriam na dostupné open-source účtovacie systémy. Kapitola je hlavne venovaná účtovaciemu systému A2Billing, ktorý patrí v súčasnej dobe medzi najpoužívanejšie a najznámejšie systémy. Tretia kapitola je venovaná samotnej praktickej realizácii, v ktorej budem navrhovať a realizovať vlastný účtovací systém na platforme OpenBTS. Využitím rôznych programovacích jazykov a softwarových nástrojov sa pokusím do tohto systému integrovať predplatené hlasové služby tak, aby svojou funkčnosťou čo najviac pripomínal prevádzku tradičného mobilného operátora. Vo štvrtej kapitole popíšem princíp prístupu k účtovacím službám, akým budú zákazníci pristupovať ku svojim kontám a akým spôsobom budú uhrádzať jednotlivé faktúry. V záverečnej časti zhodnotím dosiahnuté výsledky a funkčnosť navrhnutého systému.

2 Úvod do projektu OpenBTS

Realizácia vlastnej GSM siete prostredníctvom vlastnej základňovej stanice, ktorá je schopná vykonávať funkciu smerovania hovorov skrz celý svet a náklady na jej prevádzku by predstavovali desatinu prevádzkových nákladov súčasnej technológie už dávno nie je len predstavou o budúcnosti. Vlastnú GSM sieť môžeme realizovať prostredníctvom open-source unixovej aplikácie OpenBTS.

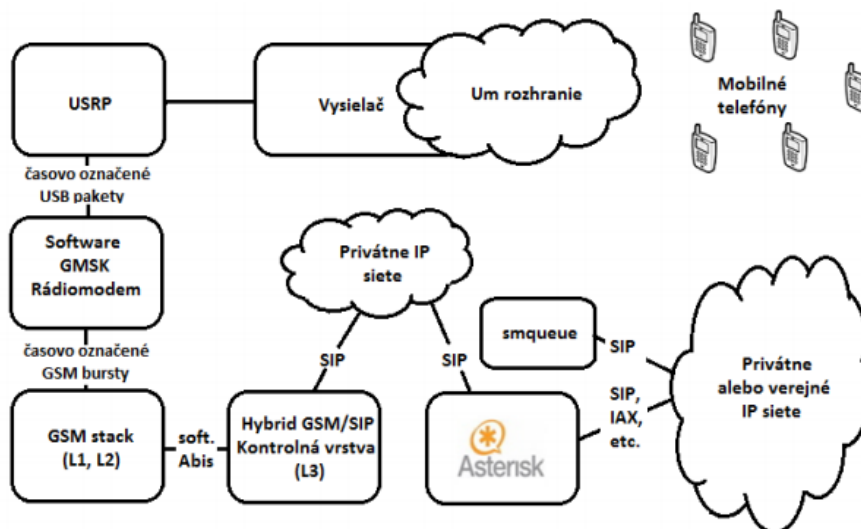
Jedná sa o systém, ktorý sa zameriava na revolúciu v mobilných sieťach nahradením starších telekomunikačných protokolov a starých hardwarových systémov prostredníctvom IP protokolu a flexibilnou softwarovou architektúrou. Oproti súčasným technológiám je omnoho menší a zároveň menej náročný na spotrebu elektrickej energie. Vďaka pripojeniu k internetu by si mohli majitelia mobilných telefónnych prístrojov navzájom telefonovať v rámci celého sveta.

Cieľom tohoto projektu je zabezpečenie nenáročného a lacnejšieho prístupu k mobilným službám v rozvojových krajinách sveta, kde infraštruktúra je nedostatočne vybudovaná alebo v obtiažne prístupných miestach ako napríklad ropné plošiny. OpenBTS implementuje prvé tri vrstvy GSM siete a zdrojový kód je napísaný v programovacom jazyku C++. Architektúra je dostupná pre inovácie od kohokoľvek, čo umožňuje vývoj nových aplikácií, služieb a výrazne zjednodušuje zakladanie a prevádzku mobilnej siete. V tejto kapitole je čerpané z nasledujúcich informačných zdrojov: [1], [2], [3], [4], [5], [6], [7].

2.1 Architektúra projektu OpenBTS

Základňová stanica BTS súčasnej GSM siete využíva pre smerovanie prichádzajúcich alebo odchádzajúcich telefónnych hovorov mobilnú rádiatelefonnu ústredňu MSC a navyše je riadená prostredníctvom základňovej riadiacej jednotky BSC. OpenBTS však na realizáciu GSM siete používa univerzálny softwarovo riadený vysielateľ USRP, poskytujúci um rozhranie pre mobilné telefóny. Funkcia smerovania a prepínania hovorov je vykonávaná softwarovou pobočkovou telefónnou ústredňou Asterisk VoIP.

OpenBTS je rozhraním medzi Asteriskom a mobilnými zariadeniami. Namiesto komunikácie s BSC zachytí IMSI mobilného telefónu a vytvorí obraz zariadenia v SIP sieti s priradeným ID. Následne sa telefón pokúša o registráciu s Asteriskom. V prípade úspešnej registrácie a pridelenia telefónneho čísla už nič nebráni tomu aby sme boli schopní komunikácie s ďalším mobilným telefónom taktiež úspešne registrovaným v telefónnom zozname Asterisku.



Obrázok 1: *Architektúra systému OpenBTS*

2.1.1 USRP (Universal Software Radio Peripheral)

Hlavným stavebným kameňom OpenBTS architektúry je univerzálny softwarovo riadený hardware USRP, ktorý vyvinula spoločnosť Ettus Research. Jedná sa o pomerne lacný hardware, ktorý je ľahko prispôsobiteľný GSM vysielateľu. Komunikácia s mobilnými zariadeniami je zabezpečená pomocou vysielania a prijímania rádiových signálov, avšak nie je možné realizovať pripojenie OpenBTS k súčasnej GSM infraštruktúre.

Systém USRP pozostáva zo základnej dosky, ktorá môže byť rozšírená o prídavné karty pre zabezpečenie vysielania a prijímania signálov v rozličných frekvenčných pásmach, USB rozhrania, cez ktoré komunikuje s počítačom a programovateľného hriadlovoho poľa FPGA. Základná doska USRP sa skladá z A/D a D/A prevodníku, rozhrania pre procesor, regulátoru energie, systému pre generovanie a synchronizáciu hodín a FPGA.

Pre správnu funkčnosť a riadenie USRP je potrebný UHD ovládač. Jedná sa o open-source ovládač, ktorý je kompatibilný so všetkými hlavnými operačnými systémami, či už Linux alebo Windows. Je ho možné použiť samostatne alebo aj s aplikáciou Labview, GNU Radio alebo už spomenutou aplikáciou OpenBTS. Využívajú ho všetky zariadenia od spoločnosti Ettus Research až na USRP1, ktorý pre svoju správnu funkčnosť potrebuje GNU rádio ovládač.

2.1.2 Vrstvový model OpenBTS

- L0 - vysielateľ
- L1(Fyzická vrstva) - funkcie FEC a TDM
- L2(Linková vrstva) - LAPDm
- L3 - funkcie pre správu signalizácie a konektivity

2.1.3 Hardwarové požiadavky

- Počítač pre prepojenie s USRP prostredníctvom USB rozhrania.
- USRP od spoločnosti Ettus Research. V tejto diplomovej práci budem pracovať s USRP N210.
- Rozširujúce dosky pre zlepšenie kvality a pokrytia signálov.
- Anténa
- 52 Mhz hodiny. Originálna USRP doska je dodávaná s 64 Mhz kryštálovým oscilátorom a niektoré telefónne prístroje by neboli schopné registrácie do OpenBTS mobilnej siete.
- Mobilný telefón
- SIM karty. Je možnosť si zakúpiť programovateľnú, ale úplne postačí aj štandardná SIM karta od mobilného operátora.



Obrázok 2: USRP N210[1]

2.1.4 Smqueue

V systéme OpenBTS funkciu zasielania textových správ z jedného mobilného telefónu do druhého a prepínanie správ plní server smqueue. Jádrom tohto serveru je tvorené frontou správ, ktoré čakajú na svoje doručenie. Jednotlivé správy sa neustále pokúšajú o svoje doručenie, vo fronte však čakajú až do tej doby, kým nie je správa potvrdená ako doručená alebo popripade označená za nedoručiteľnú. Smqueue pracuje s dvomi druhmi adries. Prvými sú numerické adresy ISDN/E.164 a druhými sú SIP užívateľské mená, ktoré nie sú zcela numerické. Smqueue v systéme OpenBTS nahradzuje a plní funkciu MSC a SMSC jednotky z tradičnej GSM siete.

2.1.5 **Subscriber registry**

Subscriber registry môžeme prirovnať konvenčnému SIP registru, avšak rozšíreným o funkcie podporujúce mobilitu a autentizáciu. Vo svojej podstate sa jedná o databázový server disponujúci rozhraním pre spracovanie metód SIP REGISTER, SS7-MAP a DIAMETER rozhraní do PLMN pre potrebu roamingu. OpenBTS komunikuje so Subscriber registry prostredníctvom SIP protokolu a ostatné sieťové komponenty prístupujú k tomuto serveru pomocou SQL príkazov. Pre obmedzenie záťaže sa do BTS jednotiek ukládajú časti SR.

2.1.6 **OpenBTS Command Line Interface (CLI)**

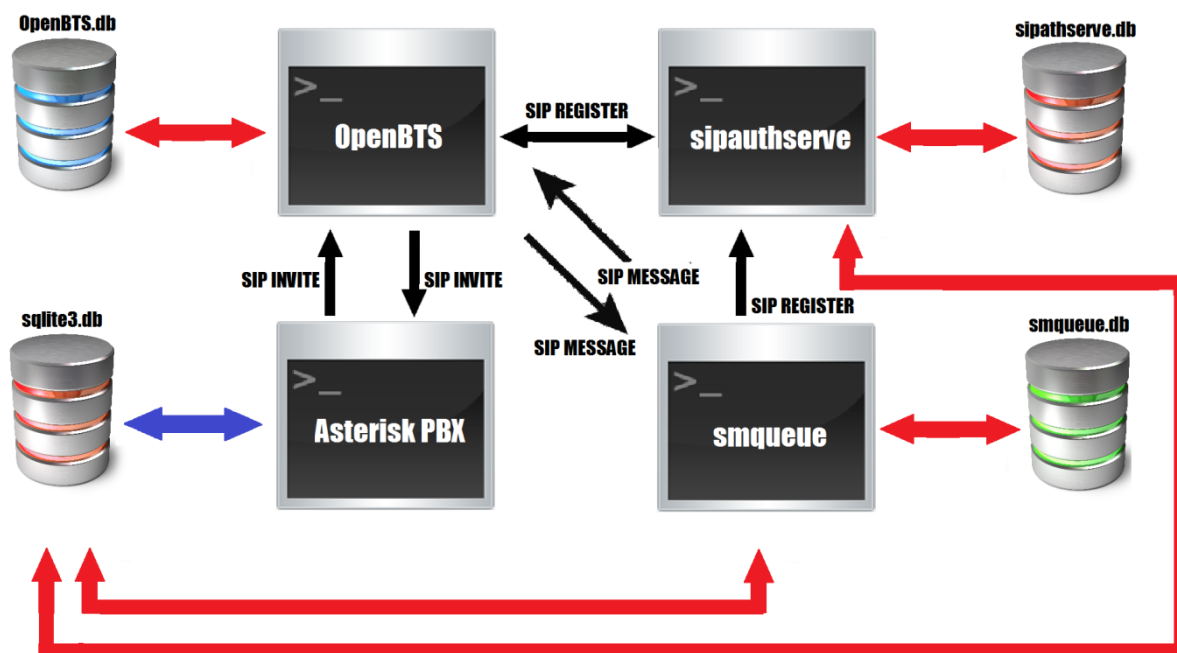
OpenBTS konzolová aplikácia sa vola CLI. CLI umožňuje monitorovanie stavu systému a zmenu mnoho OpenBTS prevádzkových parametrov a reálnom čase. Medzi dôležité príkazy OpenBTSCLI patrí:

- Config – príkaz pre zobrazenie a úpravu parametrov v konfiguračnej tabuľke
- Sendsms – zaslanie SMS textovej správy na konkrétnu mobilnú stanicu
- Tmsis – príkaz pre manipuláciu s TMSI tabuľkou a zobrazenie IMSI čísiel
- Trans – príkaz pre zobrazenie kompletnej tabuľky o transakciách
- Calls – zoznam prebiehajúcich hovorov a SMS správ
- Shutdown – vypnutie procesu OpenBTS
- Load – zoznam aktívnych kanálov
- Help – pomocník
- Endcall – ukončenie hovoru
- Cellid – zobrazenie alebo zmena cell id parametrov napr. kód krajiny (MCC), kód siete (MNC) a číslo oblasti (LAC).

2.1.7 **Databáze používané v systéme OpenBTS**

Väčšina konfiguračných parametrov zodpovedá dvojici kľúč/hodnota v databáze. V systéme OpenBTS sa nachádza niekoľko rozličných databáz, ktoré sú všetky vo formáte SQLite. Tieto databázy môže byť upravované prostredníctvom niektorého z SQLite nástrojov ako je napríklad sqliteman, použitím sqlite3 unixového príkazu alebo taktiež prostredníctvom príkazov v OpenBTSCLI klientskom rozhraní.

- Databáza OpenBTS, nachádzajúca sa v adresári /etc/OpenBTS/OpenBTS.db
- SIP autorizačná a registračná databáza sipauthserve, umiestnená v /etc/OpenBTS/sipauthserve.db
- Databáza smqueue.db, nachádzajúca sa v rovnakom adresári jako predchádzajúce dve.
- PBX databáza, ktorá je defaultne uložená v /var/lib/asterisk/sqlite3dir/sqlite3.db



Obrázok 3: Databáze a ich komunikácia s komponentami OpenBTS

2.1.8 Asterisk

Neoddeliteľnou súčasťou OpenBTS architektúry je open-source unixová aplikácia Asterisk. Vlastnosti tejto aplikácie môžeme prirovnať štandardnej telefónnej ústredne. Asterisk umožňuje VoIP, analógovú PSTN, ale aj ISDN digitálnu telefóniu prostredníctvom protokolov IAX, SIP, H.323, MGCP a SCCP. Ďalej disponuje funkciou interaktívneho hlasového systému (IVR), ktorý predstavuje hlasom alebo tónovou voľbou ovládaný terminál, ktorý zastupuje funkciu operátora prepojením hovorov a zanechaním hlasovej správy. Asterisk taktiež obsahuje funkciu automatickej distribúcie volaní (ADC), ktorá skupine terminálov dodáva prichádzajúce hovory a zabezpečuje ich smerovanie. Použitím Asterisku v OpenBTS je eliminovaná nutnosť použitia prvkov HLR a MSC z GSM systému.

2.1.9 Asterisk Management Interface

Prostredníctvom rozhrania Asterisk Management Interface (AMI) môžu externé programy komunikovať, monitorovať a riadiť Asterisk prostredníctvom TCP/IP streamu. V rámci vzdialenej komunikácie s konzolou Asterisku je možné využívať množstvo príkazov, pri ktorých je nutné dodržiavať ich formálny zápis. Úpravu konfigurácie AMI rozhrania môžeme vykonať v súbore `/etc/asterisk/manager.conf`, kde môžeme stanoviť pravidla a definovať účty pre prístup k danému rozhraniu.

AMI definuje tri základné typy správ:

- Akcia - pomocou pripojeného klienta k Asterisku sa odosiela paket, ktorý žiada o vykonanie určitej akcie. Tento paket obsahuje názov operácie, ktorá má byť vykonaná spolu so všetkými potrebnými parametrami.
- Odpoveď – odpoveď na poslednú akciu, ktorá je odoslaná Asteriskom smerom ku klientovi.
- Události – informácie generované jádrom Asterisku a poslané klientovi.

Príklady využitia AMI:

- Vytvorenie hovorového kanálu
- Sledovanie stavu hovorového kanálu
- Ukočenie prebiehajúceho hovoru
- Možnosť preddefinovať maximálnu dĺžku hovoru
- Možnosť zistenia stavu prepojenia s ostatnými telefónnymi ústredňami

2.1.10 Asterisk Gateway Interface

Pomocou komponent Asterisk Gateway Interface (AGI) môžeme s týmto rozhraním komunikovať nielen s vlastným programovacím jazykom AEL, ale môže byť rozšírený o ďalšie externé aplikácie jako napríklad skripty. Vďaka týmto skriptom môžeme do nich nasmerovať z jadra Asterisku akýkoľvek dátový prúd a spracovať ho mimo Asterisk jadro. Komunikácia je obojsmerná, čiže v prípade, že Asterisk jadro pošle nejaký druh dát do patričnej komponenty, tak môže očakávať odpoveď. Asterisk môže pracovať s nasledujúcimi programovacími jazykmi: Perl, PHP, C, Pascal, Java, PHP, Ruby, .NET alebo Haskell.

2.1.11 Protokol SIP

Najznámejším VoIP protokolom je protokol SIP, ktorý používa pre prenos signalizácie v internetovej telefónii. SIP vychádza z HTTP protokolu, je textovo orientovaný protokol a je využívaný pre zostavenie, ukončenie a modifikáciu spojenia. Pracuje na siedmej vrstve modelu RM OSI. Zabezpečuje rovnakú funkcionálnosť ako je tomu v klasických PSTN sieťach. SIP však umožňuje nenáročnú implementáciu nových služieb, ktorých implementácia v tradičných PSTN sieťach by bola oveľa náročnejšia. V kombinácii so SIP protokolom sú používané taktiež ďalšie protokoly - SDP a RTP. SDP sa používa pre popis vlastností účastníkov komunikácie a RTP sa využíva pre prenos multimediálnych dát v reálnom čase. Na transportnej vrstve sa k prenosu používa UDP protokol.

2.1.12 SIP komunikácia

Komunikácia medzi zložkami SIP protokolu prebieha prostredníctvom nasledujúcich žiadostí:

- REGISTER – žiadosť o registráciu klienta
- INVITE – žiadosť o vytvorenie spojenia alebo o zmenu parametrov už zostaveného hovoru
- ACK – potvrdenie spojenia, prijatie konečnej odpovede na správu INVITE
- BYE – ukončenie spojenia prebiehajúceho spojenia
- CANCEL – žiadosť o ukončenie práve prebiehajúcej žiadosti INVITE
- OPTIONS – informácie o serveri, zistenie podporných funkcií, detekcia vlastností koncových zariadení. Štruktúrou podobný INVITE správe, avšak spojenie nie je vytvárané.

Na vyššie uvedené SIP žiadosti je odpovedané číselnými odpoveďami:

- 1XX – informačné správy, pokračuje spracovanie SIP správy (napr. 100 Trying)
- 2XX – správa o úspešnom ukončení žiadosti (napr. 200 OK)
- 3XX – správa o presmerovaní (napr. 305 Use Proxy)
- 4XX – správa o chybe na strane klienta (napr. 403Forbidden)
- 5XX – správa o chybe na serveri, záporná odpoveď (napr. 500 Server Internal Error)
- 6XX – správa o globálnom zlyhaní, požiadavka nebola akceptovaná žiadnym serverom (napr. 606 NotAcceptable)

158.196.115.251	158.196.244.158	SIP/SDP	746 Request: INVITE sip:2102@158.196.244.158, with session description
158.196.244.158	158.196.115.251	SIP	502 Status: 100 Trying
127.0.0.1	127.0.0.1	SIP/SDP	1005 Request: INVITE sip:IMSI230025900646530@127.0.0.1:5062, with session description
127.0.0.1	127.0.0.1	SIP	436 Status: 100 Trying
127.0.0.1	127.0.0.1	SIP	436 Status: 100 Trying
127.0.0.1	127.0.0.1	SIP	436 Status: 100 Trying
127.0.0.1	127.0.0.1	SIP	437 Status: 180 Ringing
158.196.244.158	158.196.115.251	SIP	518 Status: 180 Ringing
127.0.0.1	127.0.0.1	SIP	437 Status: 180 Ringing
127.0.0.1	127.0.0.1	SIP	437 Status: 180 Ringing
127.0.0.1	127.0.0.1	SIP	437 Status: 180 Ringing
127.0.0.1	127.0.0.1	SIP	437 Status: 180 Ringing
127.0.0.1	127.0.0.1	SIP/SDP	843 Status: 200 OK, with session description
127.0.0.1	127.0.0.1	SIP	458 Request: ACK sip:IMSI230025900646530@127.0.0.1:5062
158.196.244.158	158.196.115.251	SIP/SDP	835 Status: 200 OK, with session description
158.196.115.251	158.196.244.158	SIP	409 Request: ACK sip:2102@158.196.244.158:5060
127.0.0.1	127.0.0.1	SIP	496 Request: BYE sip:2222@127.0.0.1

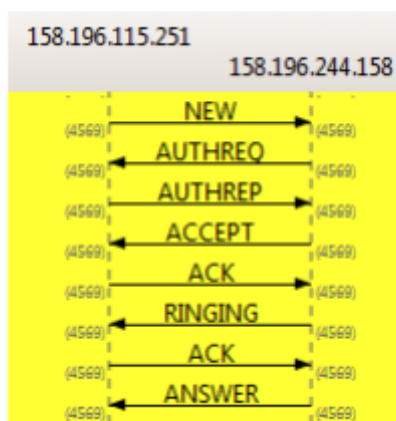
Obrázok 4: Príklad zachytenej SIP komunikácie prostredníctvom protokolového analyzátoru Wireshark[1]

2.1.13 Protokol IAX

Signalizačný protokol IAX je protokol slúžiaci ku komunikácii medzi jednotlivými Asterisk servermi. Znižuje režijné náklady na dátové spojenie, vďaka možnosti zapúzdrenia viacerých zvukových tokov do jediného paketu. IAX taktiež šetrí šírku pásma na preťaženej sieťovej linke. Od protokolu SIP sa odlišuje tým, že signalizáciu a média prenáša prostredníctvom len jedného spojenia. Ďalšou výhodou je lepší prechod cez NAT v sieti s viacerými Asterisk servermi a minimalizuje počet portov nutných k otvoreniu na firewall.

2.1.14 IAX komunikácia

- NEW – správa pre vytvorenie nového spojenia, v prípade úspechu je potvrdená správou ACCEPT.
- ACCEPT – potvrdenie zostaveného spojeni, následne sa vysiela správa ACK a volajúci čaká na správu RINGING ,ANSWER, BUSY, PROCEEDING alebo HANGUP.
- REJECT – správa informujúca, že spojenie zlyhalo, čo môže byť spôsobené buď zlým užívateľským menom, neplatnou autentizáciou alebo nekorektným heslom.
- HANGUP – správa o ukončení spojenia.
- AUTHREQ - správa vyžadujúca autentizáciu k zostaveniu spojenia.
- AUTHREP – odpoveď na správu AUTHREQ s potrebnými autentizačnými informáciami.



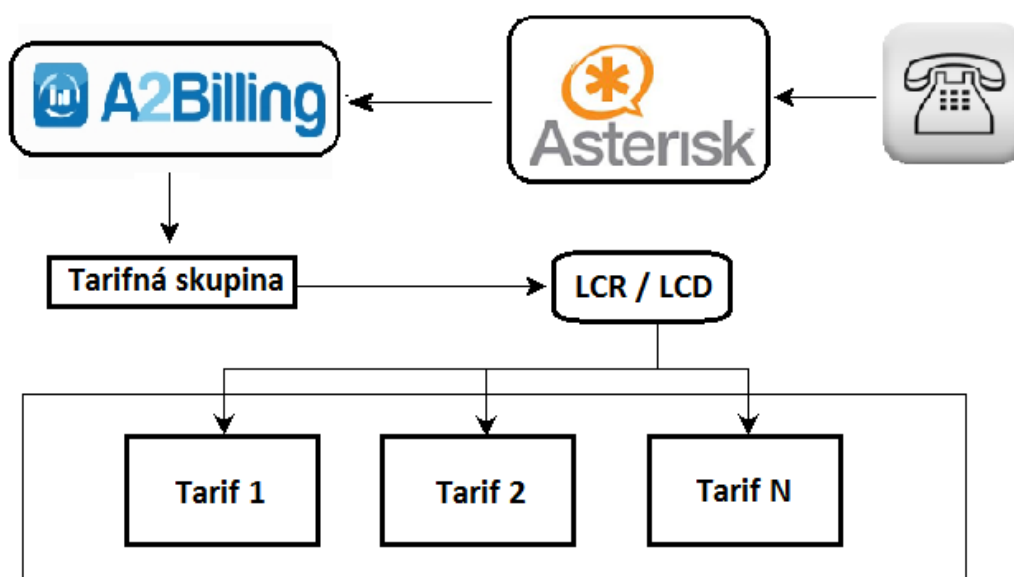
Obrázok 5: Príklad zachytenej komunikácie medzi dvoma Asterisk servermi (IAX trunk)[1]

3 Open-source účtovacie systémy

V tejto kapitole predstavím niekoľko open-source účtovacích systémov, ktoré fungujú na princípe využívania existujúcich rozšírení Asterisku a každý z nich má svoje výhody, ale aj nevýhody. Hlavne sa či už teoreticky, ale aj prakticky budem venovať účtovaciemu systému A2Billing, ktorý sa radí medzi najpoužívanejšie a najznámejšie účtovacie systémy spolupracujúce s Asteriskom. V tejto kapitole je čerpané z nasledujúcich informačných zdrojov: [9], [13], [14], [15], [16], [17].

3.1 A2Billing

Najznámejší kompletný open-source VoIP účtovací systém, ktorý zabezpečuje účtovanie hovorov v reálnom čase a je možné ho plnohodnotne využiť v rámci ľubovlného VoIP operátora. Jedná sa vlastne o externý účtovací program, ktorý je možný doinštalovať do ústredne a tým plní funkciu účtovanie hovorov. Pre správnu funkčnosť A2Billingu sa využíva AGI rozhranie Asterisku. Zakladateľom je brazílsky vývojár Aresqui Belaid. Táto účtovacia aplikácia je distribuovaná zadarmo a licencovaná General Public Licence (GPL) licenciou a je produktom spoločnosti Star2billing, ktorá sa snaží zarobiť finančné prostriedky na budúci vývoj integráciou projektu do firemných prostredí. A2billing predstavuje aplikáciu, ktorá obsahuje oprávnenia, umožňuje prihlasovanie, registráciu telefónnych užívateľov, nastavenie jednotlivých tarifkácií, fakturáciu a ešte mnoho ďalších funkcií. Tento účtovací systém pre všetky vyúčtovania hovorov v realtime režime využíva CDR záznamov, ktoré po skončení každého hovoru generuje Asterisk. A2Billing tvorí vlastne vrstvu nad Asteriskom, ako je možné vidieť na obrázku 6.



Tento obrázok popisuje vzťahy medzi jednotlivými účastníkmi a tarifami účtovania. Každý zákazník disponuje vlastnou skupinou tarifov a zároveň táto skupina môže obsahovať viac tarifov a každý tarif môže mať ešte ďalšie väzby. Pokiaľ sa v tarifnej skupine nachádza viac ako jeden tarif, jeho výber je potom riadený podľa nasledujúcich dvoch parametrov:

- Least Cost Routing (LCR) – najmenšia cena hovoru z hľadiska nákladov na strane operátora
- Least Cost Dialing (LCD) – najmenšia cena hovoru z hľadiska nákladov na strane zákazníka

3.1.1 Inštalácia účtovacieho systému A2Billing

V prvom rade je potrebné stiahnuť do svojho počítača zdrojový kód systému zo stránky github.com:

```
cd /usr/src
git clone https://github.com/Star2Billing/a2billing.git
```

Ďalej využitím nástroju pre sťahovanie dát Curl, stiahneme potrebné balíčky:

```
cd /usr/src/a2billing
curl -sS https://getcomposer.org/installer | php
php composer.phar update
php composer.phar install
```

V ďalšom inštaláčnom bode vytvorím heslo pre root mysql. Pokiaľ toto heslo bolo už vytvorené počas inštalácie Asterisku, tak je možné tento bod preskočiť.

```
mysqladmin -u root password 'mojeheslo'
```

Následne v zložke /usr/src/a2billing vytvorím A2Billing sql databázu využitím nasledujúceho príkazu:

```
cd /usr/src/a2billing
mysql -u root -p < DataBase/mysql-5.x/a2billing-createdb-user.sql
```

Po zadaní správneho root hesla bolo potrebné vyplniť tieto požadované údaje:

```
Database name: mya2billing
Database user: a2billing_ladislav
Database user password: 1234
```

Spustením skriptu install-db.sh vytvorím tabuľky a do nich vložím niektoré konfiguračné data:

```
cd /usr/src/a2billing/DataBase/mysql-5.x
./install-db.sh
```

V priebehu inštalácie som bol vyzvaný pre zadanie nasledujúcich údajov:

Database Name: mya2billing

Hostname: localhost

UserName: root

Password: mojeheslo (MySQL heslo, ktoré som vytvoril na začiatku tejto inštalácie)

Korektnú inštaláciu som overil nasledujúcim príkazom:

```
mysql -u root -p mya2billing
```

```
sql>show tables;
```

A na výstupe by sa malo objaviť, že bolo vytvorených celkovo 90 tabuliek v databázi.

```
+-----+
| Tables_in_mya2billing
| +-----+
| cc_agent          |
| cc_agent_commission |
| cc_agent_signup   |
| . . .             |
+-----+
```

```
97 rows in set (0.00 sec)
```

V adresári /usr/src/a2billing/ sa nachádza konfiguračný súbor systému A2Billing a2billing.conf, ktorý je nutné modifikovať podľa nasledujúcich riadkov:

```
cp /usr/src/a2billing/a2billing.conf /etc/
```

```
nano /etc/a2billing.conf
```

```
[database]
```

```
hostname = localhost
```

```
port = 3306
```

```
user = a2billinguser
```

```
password = a2billing
```

```
dbname = mya2billing
```

```
dbtype = mysql
```

Nasledujúcimi riadkami som zabezpečil vytvorenie niektorých súborov a pridelil im potrebné práva:

```
chmod 777 /etc/asterisk
touch /etc/asterisk/additional_a2billing_iax.conf
touch /etc/asterisk/additional_a2billing_sip.conf
echo \#include additional_a2billing_sip.conf >>
/etc/asterisk/sip_custom.conf
echo \#include additional_a2billing_iax.conf >>
/etc/asterisk/iax_custom.conf
chown -Rf asterisk:asterisk /etc/asterisk/additional_a2billing_iax.conf
chown -Rf asterisk:asterisk /etc/asterisk/additional_a2billing_sip.conf
```

Konfiguračný súbor `manager_custom.conf` umiestnený v adresári `/etc/asterisk/manager_custom.conf` som upravil podľa nasledujúceho vzoru:

```
[myasterisk]
secret=mycode
deny=0.0.0.0/0.0.0.0
permit=127.0.0.1/255.255.255.0
read=system,call,log,verbose,command,agent,user
write=system,call,log,verbose,command,agent,user
```

Ďalej nastavím niektoré grafické rozhranie a webové zložky. A2Billing disponuje tromi grafickými rozhraniami: `admin`, `agent` a `customer`.

```
mkdir /var/www/html/a2billing
cp -rf /usr/src/a2billing/admin /var/www/html/a2billing
cp -rf /usr/src/a2billing/agent /var/www/html/a2billing
cp -rf /usr/src/a2billing/customer /var/www/html/a2billing
cp -rf /usr/src/a2billing/common /var/www/html/a2billing
cp -rf /usr/src/a2billing/vendor /var/www/html/a2billing
chown -R asterisk:asterisk /var/www/html/a2billing
chmod 755 /var/www/html/a2billing/admin/templates_c
chmod 755 /var/www/html/a2billing/customer/templates_c
chmod 755 /var/www/html/a2billing/agent/templates_c
```

Následne zabezpečím zložku `/vendor`:

```
echo 'Deny from all' > /var/www/html/a2billing/vendor/.htaccess
```

Úspešnosť celej inštalácie následne overím zadaním nasledujúcej adresy do webového prehliadača:

`http://localhost/a2billing/admin`

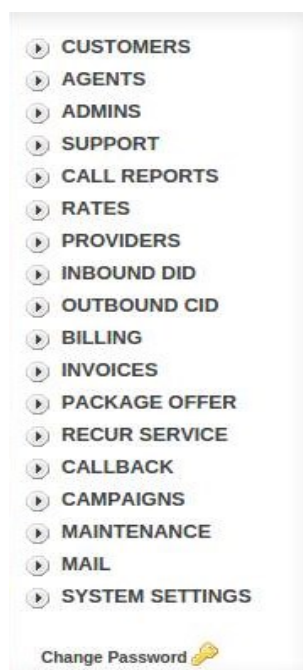
Dôkazom úspešnej inštalácie by malo byť otvorenie prihlasovacej stránky, kde po zadaní užívateľského mena a hesla, by s prístupom do administrátorského rozhrania A2Billing nemal byť žiadny problém.



Obrázok 7: Prihlasovacia stránka systému A2Billing

3.1.2 Popis základných funkcií systému A2Billing

V prípade úspešného zadania užívateľského mena a hesla, by sa malo otvoriť grafické rozhranie účtovacieho systému A2Billing, ktoré disponuje rôznymi užitočnými funkciami využiteľnými v rámci VoIP telefónie. V tejto podkapitole stručne priblížim význam jednotlivých administračných funkcií.



Obrázok 8: Základné funkcie systému A2Billing

-
- **Customers** – v tejto zložke je definovaný zoznam všetkých kariet užívateľov, ktorý boli definovaný v tejto aplikácii. Užívateľom sú tu pridelené telefónne čísla, číslo karty, stav účtu a mnoho ďalších. Taktiež je tu možné užívateľom priradiť alebo odobrať telefónne čísla, či zadefinovať telefónne čísla, ktoré môže používať. Aplikácia tiež umožňuje pridanie SIP/IAX užívateľských účtov.
 - **Billing** – systém A2Billing disponuje podporou elementárnej metódy priameho pripísania peňazí na užívateľský účet využitím množstva platobných metód. Pre dobítie svojho kreditu je možné využiť napríklad Paypal, Moneybookers či Plugnpay. Zákazník si môže zvoliť metódu platenia buď kreditnou formou alebo metódou post-pay, ktorá predstavuje platbu na konci mesiaca fakturačným spôsobom.
 - **Rates** – v tejto záložke je možné vytvoriť volací plán so skupinou vlastností spoločným pre určité tarify. Vytvárajú sa tu jednotlivé cenníky, do ktorých sú následne priradené špecifické ceny za volanie, podľa toho kam sú hovory smerované napr. vnútroštátne alebo medzinárodné hovory.
 - **Package Offer** – definovanie rôznych výhodných balíčkov, ktoré je možné priradiť zákazníkovi. Môžeme im priradiť balíčky typu free minutes, možnosť neobmedzeného tarifu do siete, kde nebude nikdy účtované hovorné alebo je možné prideliť zákazníkovi kombináciu predošlých dvoch možností.
 - **Outbound CID** – V tejto sekcii môžeme definovať smer odchádzajúcich volaní, ale aj všetky čísla, ktoré sú zaradené do rozsahu prevádzkujúceho operátora. Môžeme tu definovať skupinu, do ktorej budeme vkládať jednotlivé CID čísla. Napríklad môžeme vytvoriť skupinu v rozsahu telefónnych čísiel od 22220000 do 22229999. Prefix 2222 bude napríklad označovať, že čísla patria do moravskoslezského kraja.
 - **Providers** – Definovanie poskytovateľa, teda druhej strany, s ktorou sú uzatvorené nejaké prepojujacie dohody a jednotlivé ústredne sú prepojené využitím trunok. V tejto sekcii môžeme tieto trunky zadefinovať a modifikovať.
 - **Inbound DID** – Pomocou tejto funkcie môžeme zadávať telefónne čísla z pohľadu prichádzajúcich volaní. Jedná sa vlastne o skupinu telefónnych čísiel, ktoré budú akceptované pri prichádzajúcom hovore. Čísla, ktoré v tejto skupine nebudú zadefinované, budú ústredňou automaticky odmietnuté.
 - **Call Reports** – ako som už spomínal všetky údaje o hovoroch sú ukladané do CDR záznamov. Pomocou tejto voľby môžeme pristupovať k týmto údajom, ktoré sú spracované do prehľadnej grafickej podoby. Tieto údaje je možné taktiež na základe

rôznych parametrov filtrovať s následnou exportáciou do súboru formátu CSV alebo XML.

- **Invoices** – V tejto záložke je možné pristupovať k zoznamu všetkých faktúr v rámci systému A2Billing. Výsledné faktúry je možné exportovať do súboru vo formáte PDF. Faktúry sú generované na základe CDR záznamov a obsahujú prehľadné informácie o všetkých zákazníkových volaniach. V tejto sekcii si môžeme zobraziť prehľad o zaplatených faktúrach a taktiež o faktúrach, ktoré ešte neboli uhradené.
- **Recurr Service** – Možnosť vytvorenia a priradenia rôznych paušálnych služieb definovaných ich poskytovateľom. Môžeme tu zadať rôzne poplatky, vrátane časových intervalov, v ktorých sa majú periodicky realizovať. Taktiež môžeme nastaviť prípadne alarmy detekujúce prípadne zneužitie užívateľského účtu v dôsledku prekročenia predom nadefinovaných hodnôt. O takýchto udalostiach je užívateľ informovaný prostredníctvom emailu.
- **Callback** – Jedná sa o tzv. spätné volania. To znamená, že zákazník vytočí vopred definované číslo, ktoré je následne ústredňou akceptované a ihneď ukončené. Číslo je ústredňou zapamätané a je na neho spätne zavolané. Využitím tónovej voľby zákazník zadá externé číslo, ktoré chce vytočiť a telefónna ústredňa následne realizuje tento hovor. Výhoda tejto služby spočíva v tom, že pokiaľ užívateľ je so svojím telefónnym prístrojom v oblasti, kde hodnota tarify za odchádzajúci hovor je príliš vysoká a tak realizuje hovor využitím vlastnej ústredne, kde je definovaný nižší tarif.
- **Admins** – správa administrátorov, ktorý majú možnosť spravovať rozhranie tohto účtovacieho systému. Tiež tu môžeme zadať, do ktorých častí jednotliví administrátori budú mať povolený prístup a do ktorých mať prístup nebudú. Môžeme tu zálohovať CDR záznamy aj obsah MySQL databáze.

3.2 Freeside

Open-source tarifikačný systém s možnosťou správy účtov pre internetových poskytovateľov, ktorý teraz obsahuje aj podporu pre VoIP CDR účtovanie a predaj lístkov. Freeside podporuje Asterisk a taktiež mnoho ďalších CDR formátov. Tento systém umožňuje flexibilné určovanie cien, hodnotovacie plány, výročnú fakturáciu a zároveň umožňuje jednoduchú implementáciu nových cenníkov. V reálnom čase dokáže spracovať kreditné karty alebo elektronické šeky, či fakturácie prostredníctvom emailu alebo faxu. Súčasťou Freeside je aj kompletný, integrovaný systém predaja lístkov, kedy automaticky reaguje na email zákazníka a priradí mu číslo vstupenky.

3.3 EasyITSP

EasyITSP je open-source rezidentný alebo hosťujúci poskytovateľ internetovej telefónie, ktorý tvorí vrstvu nad pobočkovou ústredňou Asterisk. Tento software umožňuje ľuďom s menšími skúsenosťami jednoducho spustiť a riadiť služby malej pobočkovej ústredne a využívať administrátorské a zákaznícke portály. Systém je riadený prostredníctvom Asterisk AGI súborov. EasyITSP je rýchlo vyvíjajúca aplikácia na svojom trhu vďaka konštantnému vývoju nových vlastností a funkcií medzi ktoré patria napríklad:

- Monitoring Asterisku a Linuxu
- CDR reporty
- Administrátorský portál
- Zákaznícky portál
- Smerovanie hovorov
- Blokovanie hovorov
- Hlasové správy
- Music on hold
- Interaktívna hlasová odozva (IVR)

3.4 astCDRview

Open source jednoduchý software vyvíjaný pod licenciou GNU/GPL. AstCDRview je veľmi jednoduchý webovo založený mnohojazyčný Asterisk SQLite CDR prehliadač. Obsahuje základné funkcie ako monitorovanie prichádzajúcich alebo odchádzajúcich hovorov, účtovanie hovorov alebo zoznam kontaktov

3.5 AstPP

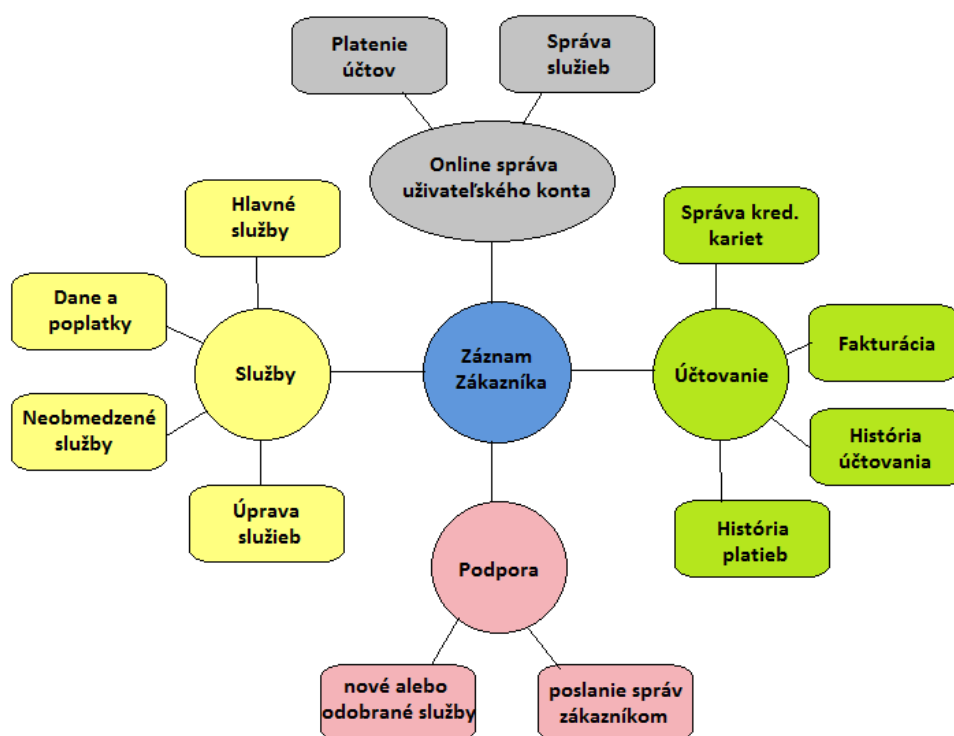
AstPP predstavuje open source webové rozhranie pre Asterisk. Tak isto obsahuje GUI editor pre niektoré zariadenia a dialplán. Administrátor v tejto aplikácii môže spravovať celý systém a zároveň určiť viacerých adminov či subadminov. Aplikácia poskytuje reporty, klientský portál alebo faktúry, ktoré je možné exportovať do CSV. Platenie účtov umožňuje prostredníctvom internetového portálu Paypalu.

3.6 CitrusDB

CitrusDB je open-source zákaznícky servis a účtovací systém. Využíva ho množstvo poskytovateľov internetových a telekomunikačných služieb. Pre svoju správnu funkčnosť potrebuje

PHP 5.2 a vyžaduje MySQL verziu 5 a vyššie. Software je licencovaný pod GPL verzia 2. Základnými rysmi softwaru sú:

- CRM, podpora lístkov, faktúrovanie, kreditné karty
- Neobmedzené množstvo užívateľov alebo zákazníkov
- Neobmedzené množstvo služieb a atribútov
- Zabezpečené uchovanie informácií o kreditných kartách pomocou OpenPGP
- Vyhýba sa cloudovým dátám



Obrázok 9: Diagram funkčnosti softwaru CitrusDB

3.7 ASTERISELL

Ďalším open source účtovacím systémom pre VoIP poskytovateľov, ktorý slúži na prezeranie, hodnotenie a účtovanie hovorov je ASTERISELL. Táto aplikácia je implementovaná prostredníctvom PHP programovacieho jazyka a je možné tento zdrojový kód upravovať, aby vyhovoval vlastným podmienkam užívateľa. Disponuje možnosťou importu CDR záznamov z lokálnych alebo aj vzdialených tabuliek, CSV súborov alebo iných logovacích súborov v rôznych formátoch. Umožňuje aplikovať rôzne druhy tarifácie od jednorázových až po mesačné faktúry. Podporuje odchádzajúce,

prichádzajúce, interné a systémové volanie. Podrobné záznamy o histórii volaní dokáže exportovať do CSV alebo Excel súborov a taktiež umožňuje ich následnú filtráciu napríklad podľa volaného čísla alebo dátumu. Pre svoju správnu funkčnosť požaduje webový server Apache, PHP, MySQL a samozrejme pobočkovú ústredňu Asterisk.

4 Návrh a realizácia účtovania služieb

V tejto kapitole sa budem dôkladne venovať samotnému návrhu a následnej realizácii vlastného účtovacieho systému na platforme openBTS a popíšem všetky potrebné náležitosti, ktoré som využil pre vytvorenie tohto systému. Pri implementácii som si musel hlavne položiť dve základné otázky. Prvou je akým spôsobom budem účtovať a monitorovať jednotlivé hovory, poprípade sms správy a druhým problémom, ktorým sa bolo potrebné zaoberať je kontrola hovoru, či náhodou počas neho nedošlo k prečerpaniu zákazníkových predplatených služieb. V tejto kapitole je čerpané z nasledujúcich informačných zdrojov: [10], [11], [12], [13], [14], [15].

4.1 Inštalácia OpenBTS a jeho súčastí

V prvom rade si stiahneme najnovšiu verziu OpenBTS zadaním nasledujúceho príkazu v termináli:

```
svn co http://wush.net/svn/range/software/public
```

OpenBTS ku správnej funkčnosti potrebuje komunikovať s USRP za pomoci ovládača UHD (USRP Hardware Driver). Najskôr však nainštalujeme všetky nevyhnutné závislosti a následne spomínaný ovládač.

```
sudo apt-get install autoconf libtool libosip2-dev libortp-dev libusb-1.0-0-dev g++ sqlite3 libsqlite3-dev erlang libreadline6-dev libncurses5-dev
```

Taktiež je potrebné nainštalovať liba53, čo je súčasťou distribúcie. Zadaním nasledujúcich príkazov by malo dojsť ku korektnej inštalácii priamo z OPENBTS_ROOT:

```
cd a53/trunk
sudo make install
```

Následne si stiahneme binárny súbor s najnovšiou verziou UHD a hneď potom tento ovládač nainštalujeme.

```
bash -c 'echo "deb
http://files.ettus.com/binaries/uhd_stable/repo/uhd/ubuntu/\
`lsb_release -cs` `lsb_release -cs` main" > \
/etc/apt/sources.list.d/ettus.list'
apt-get update
apt-get install -t `lsb_release -cs` uhd
```

Pre podporu UHD zariadení, uskutočníme následovnú konfiguráciu:

```
cd openbts/trunk
autoreconf -i
./configure --with-uhd
Make
```

Ďalej je potrebné vytvoriť symbolický odkaz vysielača potrebného pre hardware USRP:

```
 #(from OpenBTS root)
 cd apps
 ln -s ../Transceiver52M/transceiver .
```

Treba si všimnúť, že oproti minulej poslednej inštalácii boli odstránené možnosti `--resamp` a `--extref`. Výber vzorkovacej frekvencie a prevzorkovanie sa riešia automaticky prostredníctvom vysielača. Odkazovanie na externé hodiny sa povoľuje cez konfiguračnú databázu. Pre povolenie externej 10 Mhz referencie , príkaz `TRX.Reference` môže byť pridaný do konfigurácie `OpenBTS.db` s hodnotu nastavenou na 1.

```
"TRX.Reference" '0' for Internal or '1' for External 10 MHz reference
```

Ďalej potrebujeme nakonfigurovať `OpenBTS` aby fungovala korektne. Pre dosiahnutie správnej funkčnosti musím vytvoriť dva kľúčové súbory. Vytvorím si zložku `/etc/OpenBTS`, ktorá pravdepodobne pri inštalácii nebola vytvorená a vytvorím v nej súbor `OpenBTS.db` prostredníctvom nasledujúcich príkazov.

```
(from the OpenBTS directory)
 sudo mkdir /etc/OpenBTS
 sudo sqlite3 -init ./apps/OpenBTS.example.sql /etc/OpenBTS/OpenBTS.db
 ".quit"
```

Následne môžem overiť či sa databáza správne vytvorila pomocou príkazu uvedeného nižšie. Na obrazovke by sa mal zjaviť výpis s množstvom konfiguračných premenných, čo značí, že databáza bola vytvorená úspešne.

```
sqlite3 /etc/OpenBTS/OpenBTS.db .dump
```

V tomto bode je už možné skontrolovať, či `openBTS` funguje podľa predstáv.

```
(from OpenBTS root)
 cd apps
 screen -S openbts
 sudo ./OpenBTS
```

A na výstupe sa musí zjaviť nasledujúce potvrdenie o tom, že systém je pripravený na použitie:

```
system ready
use the OpenBTSCLI utility to access CLI
```

Teraz už je možné v nastavení telefónu vyhľadať testovaciu sieť `openBTS` pod názvom `00101`. Pri pokuse o pripojenie systém neprijme žiadosť a telefón odmietne. Je to z toho dôvodu, že v novej

verzii openBTS umožňuje pripojenie k sieti iba registrovaným telefónnym zariadeniam. Preto je potrebné zmeniť niektoré konfiguračné premenné. OpenBTS disponuje vlastným príkazovým riadkom, ktorý si spustíme pomocou daného príkazu:

```
(from OpenBTS root)
cd apps
screen -S btscli
sudo ./OpenBTSLI
```

Po úspešnom spustení príkazového riadku zadáme do neho príkaz `config`. Zobrazí sa množstvo konfiguračných údajov. Mňa však zaujímajú iba nasledujúce tri.

- `GSM.Radio.Band` – nastavím GSM pásmo na 900.
- `GSM.Radio.CO` - tento parameter predstavuje ARFCN, ktorému pridám hodnotu 1.
- `Control.LUR.OpenRegistration` – Nastavím hodnotu tohto parametru na regulárny výraz „.*“. Touto úpravou zabezpečím, aby OpenBTS nedomietla mobilný telefón pri jeho pokuse o registráciu.

V tomto momente už je moje BTS stanica plne funkčná. Ďalej však potrebujem dokončiť konfiguráciu niektorých komponent. A to HLR v podobe `sipauthserve`, centrum krátkych textových správ `smqueue` a pobočkovú ústredne Asterisk aby som mohol uskutočňovať hovory v rámci siete.

Pre nastavenie Subscriber Registry databáze musím najskôr vytvoriť cestu k zložke kde bude vytvorená databáza:

```
sudo mkdir -p /var/lib/asterisk/sqlite3dir
```

Následne skompilujem kód pre vytvorenie `sipauthserve`:

```
cd subscriberRegistry/trunk
make
```

Prostredníctvom nasledujúceho príkazu nakonfigurujem `sipauthserve`, za predpokladu, že zložka `/etc/OpenBTS/` bola už správne vytvorená.

```
(from subscriberRegistry root)
sudo sqlite3 -init subscriberRegistry.example.sql
/etc/OpenBTS/sipauthserve.db ".quit"
```

Teraz už nič nebráni spusteniu `sipauthserve`.

```
(from subscriberRegistry root)
Screen -S sipauthserve
sudo ./sipauthserve
```

Správnu funkčnosť sipauthserve by mal demonštrovať nasledujúci výpis na obrazovke terminálu.

```
ALERT 139639310980928 sipauthserve.cpp:214:main: ./sipauthserve
(re)starting
```

Poslednou prekážkou pred spustením plne funkčného systému je konfigurácia a spustenie smqueue, ktorého úlohou je ukladať a zároveň smerovať textové správy v sieti OpenBTS. V zložke /smqueue/trunk/ spustíme príkazy:

```
autoreconf -i
./configure
Make
```

Smqueue taktiež obsahuje konfiguračný súbor uložený v /etc/OpenBTS/smqueue.db. Nasledujúcim príkazom nastavíme parametre databáze na defaultné hodnoty. Tieto konfiguračné premenné by mali fungovať bez akejkoľvek modifikácie.

```
(from the smqueue directory)
sudo sqlite3 -init smqueue/smqueue.example.sql /etc/OpenBTS/smqueue.db
".quit"
```

Po úspešnom vykonaní predchádzajúcich konfigurácií už môžem spustiť aj smqueue.

```
(from the smqueue directory)
cd smqueue
screen -S smqueue
sudo ./smqueue
```

Po spustení sa na obrazovke musí objaviť nasledujúci výpis.

```
ALERT 140545832068928 smqueue.cpp:2421:main: smqueue (re)starting
smqueue logs to syslogd facility LOCAL7, so there's not much to see here
```

OpenBTS loguje do súboru syslogd a všetkú činnosť openbts môžem sledovať zadaním príkazu:

```
tail -f /var/log/syslog | grep OpenBTS
```

4.2 Inštalácia a konfigurácia Asterisk PBX

Ako som spomínal v prvej kapitole, tak neoddeliteľnou súčasťou správnej funkčnosti OpenBTS systému je práve softwarovo riešená telefónna ústredňa Asterisk. Jej inštaláciu vykonáme podľa nasledujúcich krokov.

Nasledujúcim príkazom najskôr nainštalujem základne závislosti pre korektnú funkčnosť Asterisku:

```
apt-get install build-essential wget libssl-dev libncurses5-dev libnewt-  
dev libxml2-dev linux-headers-$(uname -r) libsqlite3-dev uuid-dev
```

Ďalším príkazom zabezpečím stiahnutie aktuálnej verzie Asterisku:

```
wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-11-  
current.tar.gz
```

Stiahnutý archív rozbalíme a následne spustíme inštaláciu systému:

```
tar zxvf asterisk*  
cd /usr/src/asterisk*  
./configure && make menuselect && make && make install && make config &&  
make samples
```

Nasledujúcimi príkazmi už môžem Asterik uviesť do prevádzky:

```
/etc/init.d/asterisk start  
screen -S asterisk  
asterisk -rvvv
```

Po úspešnej inštalácii Asterisku je potrebné upravenie niektorých jeho konfiguračných súborov. Najskôr som súbor `/etc/odbcinst.ini` aktualizoval o následovnú konfiguráciu.

```
[SQLite3]  
Description=SQLite3 ODBC Driver  
Driver=/usr/local/lib/libsqlite3odbc.so  
Setup=/usr/local/lib/libsqlite3odbc.so  
Threading=2
```

V súbore `/etc/odbc.ini` je potrebné upraviť sekciu `[asterisk]`.

```
[asterisk]  
Description=SQLite3 database  
Driver=SQLite3  
Database=/var/lib/asterisk/sqlite3dir/sqlite3.db # optional lock timeout in  
milliseconds  
Timeout=2000
```

Pre súbory `odbc.ini` a `odbcinst.ini` boli pridané symbolické odkazy.

```
root@openbts:/# cd /usr/local/etc  
root@openbts:/# ln -s /etc/odbc.ini  
root@openbts:/# ln -s /etc/odbcinst.ini  
root@openbts:/# cd /root  
root@openbts:/# ln -s /etc/odbc.ini .odbc.ini
```

```
root@openbts:/# ln -s /etc/odbcinst.ini .odbcinst.ini
```

Modifikáciou konfiguračného súboru `/etc/asterisk/modules.conf` zabezpečím povolenie pre automatické nahrávanie modulov.

```
[modules]
autoload=yes
preload => res_config_odbc.so
; noload => res_config_odbc.so
```

Následnou úpravou súboru `/etc/asterisk/extconfig.conf` povolím Asterisku používanie odbc databáze v real time móde.

```
[settings]
sipusers => odbc,asterisk,sip_buddies
sippeers => odbc,asterisk,sip_buddies
```

Konfiguračný súbor `/etc/asterisk/res_odbc.conf` som upravil podľa nasledujúcich riadkov:

```
[asterisk]
enabled => yes
dsn => asterisk
pre-connect => yes
```

V súbore `/etc/asterisk/func_odbc.conf` som povolil použitie SQL príkazov v dialpláne.

```
[SQL]
dsn=asterisk
readsql=${ARG1}
```

Nakoniec som už len modifikoval dialplán Asterisku kde som pridal context `[phones]` s následovnou konfiguráciou, ktorá pri volaní mobilných staníc v rámci OpenBTS siete umožňuje vyhľadávanie záznamov v sqlite3 databázi Asterisku.

```
[phones]
exten => _N.,1,Set(Name=${ODBC_SQL(select dial from dialdata_table where
exten = \"${EXTEN}\")})
exten => _N.,n,GotoIf("${Name}" = "" ?outbound-trunk,${EXTEN},1)
exten => _N.,n,Set(IPAddr=${ODBC_SQL(select ipaddr from sip_buddies where
name = \"${Name}\")})
exten => _N.,n,GotoIf("${IPAddr}" = "" ?outbound-trunk,${EXTEN},1)
exten => _N.,n,Dial(SIP/${Name}@${IPAddr}:5062)
```

4.3 Ďalšie použité technológie

Pri realizácii svojej diplomovej práce som využil množstvo ďalších nástrojov ako je webový server Apache, skriptovací jazyk PHP, nástroj pre ovládanie MySQL databáze phpMyAdmin, knižnicu JpGraph pre vytváranie grafov a nástroj FPDF pre vytváranie PDF súborov využitím PHP. V nasledujúcich riadkoch popisujem základné vlastnosti jednotlivých nástrojov a oboznamujem čitateľa o ich spôsobe inštalácie a následovnej konfigurácie.

4.3.1 Apache

Server Apache je najrozšírenejší multiplatformný HTTP softwarový webový server, ktorý je v súčasnosti využívaný na väčšine serverov. Je prispôsobový rôznym potrebám a požiadavkám užívateľa vďaka jeho modulovej architektúre. Dostupný je pre Windows platformy, ale aj taktiež aj pre platformy Linuxové a spolupracuje s rôznymi skriptovacími jazykmi ako je Perl, Python alebo PHP. Medzi jeho hlavné funkcie patrí napríklad:

- Zaistenie zabezpečeného spojenia HTTP
- Podpora autentizácie užívateľov
- Spolupráca so skriptovacími jazykmi
- Virtual hosting – možnosť aplikácie viacerých domén
- Podpora reverzného proxy – používané na serverových farmách pre load balancing

Inštalácia Apache nie je vôbec obtiažna nakoľko tento server je dostupný ako Ubuntu balíček, takže jeho pre jeho inštaláciu stačí spustiť následovný príkaz v termináli:

```
apt-get install apache2
```

4.3.2 PHP

PHP predstavuje skriptovací programovací jazyk, ktorý sa využíva hlavne pri tvorení webových aplikácií či dynamických webových stránok súčasne s jazykmi HTML, XHTML či XML. Jedná sa o interpretovaný jazyk, keďže všetky jeho skripty najskôr spracováva serverová strana a až následne je s výsledkom oboznámený užívateľ. Funkcie PHP však nie sú závislé len na serveroch typu HTML, ale je možné prevádzkovať programy ako konzolovú aplikáciu alebo ako samostatnú GUI.

Disponuje množstvom užitočných knižníc využívaných pri spracovaní textu, prístupu alebo pri práci so súborami. Taktiež sa používa pri pripojení k niektorým databázam, najčastejšie sa však PHP jazyk využíva pri práci s MySQL databázami. PHP podporuje tiež množstvo internetových protokolov ako napr. SMTP, FTP, HTTP, POP3 alebo IMAP.

Inštalácia PHP5 pre spoluprácu s webovým serverom Apache sa vykoná nasledujúcim zadaním príkazu do linuxového terminálu:

```
sudo apt-get install php5 libapache2-mod-php5
```

4.3.3 MySQL

Pod pojmom MySQL rozumieme relačný databázový systém, za ktorého vznikom stojí spoločnosť MySQL AB. Jedná sa o databázový systém kompatibilný s rôznymi platformami, čiže je možné ho prevádzkovať na rozličných operačných systémoch. Najčastejšie je používaným s kombináciou softwarov tzv. LAMP (Linux, Apache, MySQL, PHP) . V tejto diplomovej práci sú databázy neoddeliteľnou súčasťou, v ktorých sú ukládané množstvá dôležitých dát.

MySQL nainštalujeme prostredníctvom nasledujúceho príkazu v prostredí Ubuntu:

```
apt-get install mysql-server mysql-client
```

4.3.4 phpMyAdmin

Táto aplikácia umožňuje kompletne ovládanie MySQL databázového serveru cez webové rozhranie. Ku svojej funkcii využíva skriptovací jazyk PHP a webový server, vďaka čomu je prostredníctvom neho možné manipulovať a spravovať MySQL databázy odkiaľkoľvek. Inštaláciu spustíme nasledovným príkazom:

```
apt-get install phpmyadmin
```

4.3.5 PHP JpGraph

JpGraph predstavuje objektovo-orientovanú knižnicu pre skriptovací jazyk PHP. Prostredníctvom tejto knižnice napísanej v PHP je možné vytvárať grafy v PHP skriptoch. V tejto diplomovej práci využívam tento nástroj pre vytváranie grafových štatistík ohľadom počtu hovorov, smsiek alebo celkové množstvo peňazí investovaných zákazníkom do hlasových služieb za mesačné obdobia.

Inštaláciu tohto projektu začneme stiahnutím aktuálneho balíčku z oficiálnej stránky projektu. Následne tento balíček rozbalíme využitím príkazu:

```
tar -zxvf jpgraph-3.5.0b1.tar.gz
```

Ďalej adresár src skopírujem do adresára kde sa nachádzajú webové stránky môjho OpenBTS projektu a následne sa príkazom cd dostanem do tejto zložky, kde premenujem src zložku na jpgraph a pomocou príkazu chmod 777 jej nastavím maximálne práve pre čítanie, zápis aj spúšťanie:

```
sudo cp jpgraph-3.5.0b1/src/ /var/www/my_project/  
cd /var/www/my_project  
sudo chmod 777 jpgraph -R
```

4.3.6 FPDF

FPDF predstavuje triedu PHP, ktorá umožňuje generovanie PDF súborov bez využitia PDFlib knižnice. Je voľne stiahnuteľná a prispôsobivá rôznym požiadavkám užívateľa. Medzi ďalšie hlavné výhody tejto knižnice patrí:

- voľba meracej jednotky, formátu stránky a nastavenia okrajov
- prispôsobenie záhlaví a zápäť
- automatický zlom strany
- automatický koniec riadku a zarovnanie textu
- podpora obrázkových formátov JPEG, PNG a GIF
- voľba farby
- kompresia stránok

Vo svojej diplomovej práci tento nástroj využijem pri vytváraní pdf súborov s prehľadom o fakturačných údajoch o hovoroch, ktoré si môže zákazník stiahnuť prihlásením sa ku svojmu OpenBTS účtu. Pre jeho inštaláciu je potrebné spustiť nasledovný príkaz:

```
apt-get install php-fpdf
```

Po úspešnej inštalácii vytvorím symbolický odkaz v adresári, kde sa nachádzajú stránky OpenBTS operátora a reštartujem webový server Apache:

```
ln -s /usr/share/php/fpdf /var/www/fpdf  
/etc/init.d/apache2 restart
```

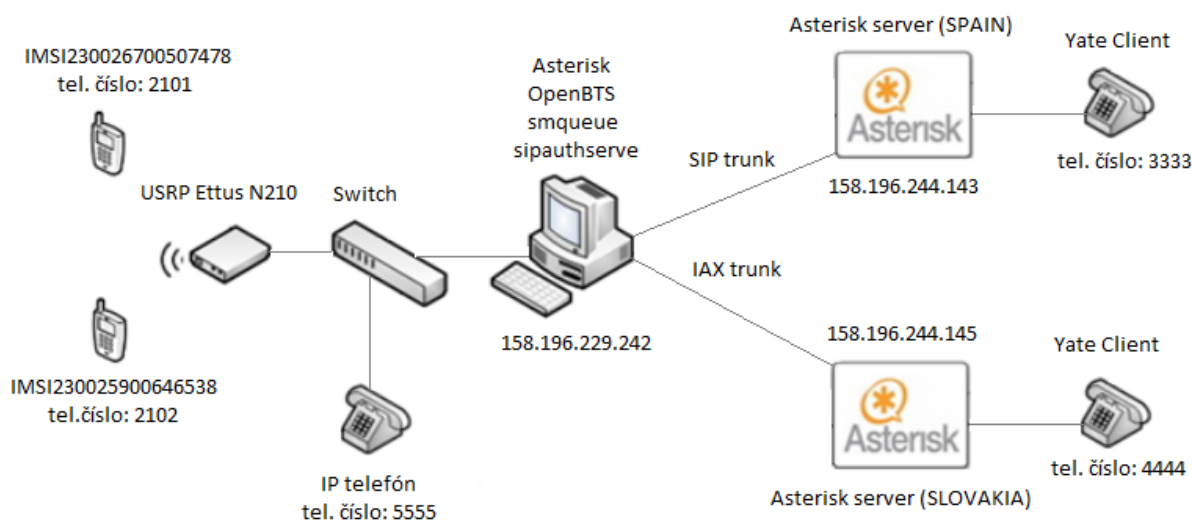
Pre vytváranie PDF využitím PHP už je len potrebné aby zdrojový kód obsahoval nasledujúce riadky:

```
<?php  
require (' fpdf/fpdf.php' );  
?>
```

4.4 Automatická registrácia IMSI čísiel v sieti OpenBTS

Na obrázku 10 je možné vidieť schému zapojenia použitéj pri realizácii tejto diplomovej práce. Na počítači s pridelenou IP adresou 158.196.229.242 sú spustené paralelne program OpenBTS, Asterisk, smqueue a sipauthserve. Prostredníctvom ethernet káblu je PC spojený so sieťovým prepínačom (switch), ku ktorému sú ďalej pripojené s USRP N210 a IP telefón. Pobočková ústredňa Asterisk je spojená prostredníctvom SIP trunku s ďalším Asterisk serverom s IP adresou

158.196.244.143 a zároveň spojenie s druhým Asterisk serverom s IP 158.196.244.145 je realizované prostredníctvom IAX trunku. K jednotlivým serverom sú pripojené softwarové telefóny Yate Client.



Obrázok 10: Schéma zapojenia

Vo svojej bakalárskej práci som implementoval program v jazyku C++ (obrázok 11), ktorý po manuálnom zadaní IMSI čísla telefónu ako vstupného parametru upravil obidva konfiguračné súbory Asterisku a jeho tabuľky `dialdata_table` a `sip_buddies` v SQL databáze. Následne po vytočení telefónneho čísla už bola realizovateľná komunikácia medzi mobilnými zariadeniami v sieti OpenBTS.

```
root@ladis-Lenovo-G570:~/home/ladis/Desktop/dist# ./astprg IMSI230025900646539

Opening database...OK
IMSI not used, creating new record.
Inserting to db: id(4), number(2103), imsi(IMSI230025900646539)
Inserting to dialdata_table...OK
Inserting to SIP_BUDDIES...OK
/etc/asterisk/sip.conf updated.
/etc/asterisk/extensions.conf updated.
Closing DB...OK
```

Obrázok 11: Manuálna registrácia IMSI

V rámci diplomovej práce som však tento program vylepšil a momentálne je tento proces plne automatizovaný. V praxi to znamená, že akonáhle sa mobilná stanica úspešne prihlási do siete, tak je okamžite jeho IMSI číslo uložené do databáze `TMSITable.db`, ktorá sa nachádza v adresári `/var/run/OpenBTS/`. Túto databázu prostredníctvom programu `openbts_reg.cpp` (obrázok 12)

neustále kontrolujem, či sa v nej objavilo nové IMSI číslo, čo by znamenalo, že sa do siete pripojil nový záujemca o služby operátora OpenBTS. IMSI číslo je následne porovnané so sip_buddies tabuľkou, či už sa v nej náhodou nenachádza. V prípade, ak už má záznam v tejto tabuľke, tak je ignorované, v opačnom prípade mu program využitím funkcie generatePhoneNumber() prideli telefónne číslo. Využitím funkcií insertToDialData() a insertToSipBuddies() upraví sqlite3 databázu Asterisku. Následne už ostáva len vložiť IMSI spolu s patričnými parametrami do konfiguračných súborov pobočkovej ústredne. Úpravu konfiguračného súboru sip.conf zabezpečuje funkcia createRecordInSIPConf(), ktorá do tohto súboru vloží nasledovnú konfiguráciu:

```
[IMSI230024701231666]
callerid=2101
type=friend
host=dynamic
canreinvite=no
context=remote
allow=gsm
dtmfmode=RFC2833
```

Funkcia createRecordInExtensionsConf() zabezpečí vloženie konfigurácie do kontextu [internal] v súbore extensions.conf.

```
[external]
exten => 2101,1,Dial(SIP/ IMSI230024701231666@127.0.0.1:5062)
```

Pridanie vyššie uvedeného riadku umožňuje inej mobilnej stanici registrovanej v tejto sieti dovolať sa na toto mobilné zariadenie vytočením čísla 2101.

```
INFO: Searching for new IMSI...
SUCCESS: New IMSI found =>IMSI230024701231666 !!!
INFO: Opening database...OK
INFO: IMSI not used, creating new record.
INFO: Inserting to db: id(2), number(2101), imsi(IMSI230024701231666)
SUCCESS: Inserting to dialdata_table...OK
SUCCESS: Inserting to SIP_BUDDIES...OK
SUCCESS: /etc/asterisk/sip.conf updated.
SUCCESS: /etc/asterisk/extensions.conf updated.
INFO: Closing DB...OK
INFO: Welcome message was successfully sent !!

INFO: Searching for new IMSI...
```

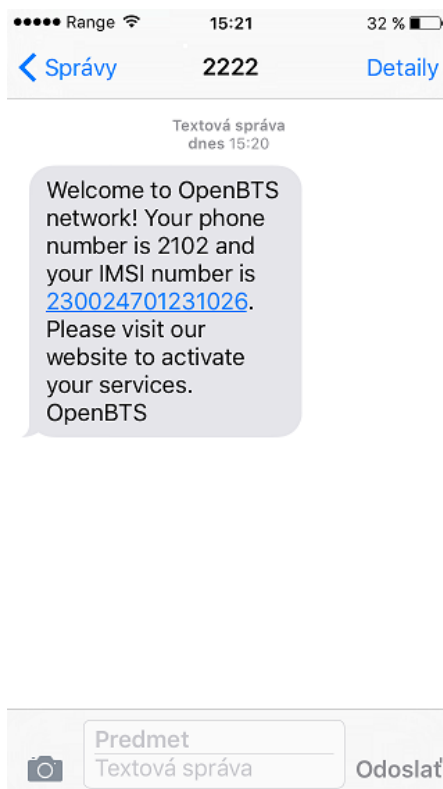
Obrázok 12: Automatická registrácia v sieti openBTS

Využitím funkcie `insertLoginMySQL()` je telefónne číslo, IMSI číslo a automaticky vygenerované ID číslo záznamu vložené do MySQL databáze. Štruktúru tejto databáze je možné vidieť na nasledujúcej tabuľke. Username predstavuje telefónne číslo užívateľa a password je IMSI číslo jeho mobilného zariadenia.

Tabuľka 1: štruktúra MySQL databáze *users.db*

Názov atribútu	Dátový typ
ID	int(11)
Username	varchar(80)
Password	varchar(80)

Ako je možné vidieť na obrázku 9, tak novému zákazníkovi je následne automaticky poslaná sms správa o pridelení telefónneho čísla a hodnota IMSI čísla jeho SIM karty. Tieto informácie následne využije ako prihlasovacie údaje ku svojmu účtu na webovej stránke poskytovateľa služieb, kde si môže aktivovať hlasové služby. Každé telefónne číslo je iné a zároveň IMSI číslo každej užívateľskej SIM karty je rozdielne, takže užívateľské meno a heslo dvoch užívateľov nebudú za žiadných okolností rovnaké. Tento prístup k riadeniu služieb na základe účtovania je popísaný v samostatnej piatej kapitole tejto diplomovej práce.



Obrázok 13: SMS správa novému zákazníkovi

4.5 Spôsob účtovania za hlasové služby v sieti OpenBTS

Nakoľko hovory v sieti OpenBTS sú smerované cez pobočkovú ústredňu Asterisk, tak tá zaznamenáva podrobné dátové záznamy, ktoré obsahujú informácie o všetkých realizovaných telefónnych hovoroch. Tieto záznamy o volaní alebo „calling data records (CDR)“ sú po každom hovore Asteriskom spracované a zapísané do súboru typu CSV uloženého v adresári `/var/log/asterisk/cdr-csv/Master.csv`. CDR záznamy bývajú defaultne deaktivované a pre ich aktiváciu je potrebné v príkazovom riadku Asterisku zistiť prítomnosť modulu `cdr_csv.so` prostredníctvom príkazu „`module show like cdr`“. Pokiaľ daný modul vo vypísanom zozname modulov nie je vidno, tak je pre jeho aktiváciu nutné zadať príkat „`module load cdr_csv.so`“. Lepšou možnosťou je upraviť konfiguračný súbor `modules.conf` v adresári `/etc/asterisk/`, kde je potreba doplniť riadok „`load cdr_csv`“ a následne asterisk reštartovať.

CDR záznamy obsahujú informácie pre spracovanie, ktoré sú veľmi dôležité pri realizácii vlastného účtovacieho systému. Dôležité informácie, ktoré budem spracovávať z CDR záznamov sú:

- Číslo volajúceho
- Číslo volaného účastníka
- Čas začiatku hovoru
- Čas ukončenia hovoru
- Použitý trunk
- Použitý kontext
- Dátum a čas
- Údaj či bol hovor prijatý volanou stranou

Typický CDR záznam vypadá nasledovne:

```
, "2101", "1003", "SPAIN", "2102", "SIP/2101-00000006", "SIP/2102-00000007", "Dial", "SIP/2101", "2016-03-13 16:56:18", "2016-03-16 16:56:26", "2013-11-13 16:51:22", 11, 8, "ANSWERED", "DOCUMENTATION", "1384361471.6", ""
```

Z tohto záznamu môžem vyčítať, že užívateľ s telefónnym číslom 2101 vytočil na svojom telefónnom prístroji číslo 2102, ktoré sa v dialpláne nachádza v kontexte SPAIN. Hovor bol akceptovaný, čo nám dokazuje vo výpise položka ANSWERED a bol uskutočnený cez SIP trunk. Účastník s číslom 2102 hovor prijal v čase 16:56:18 a bol ukončený 16:56:26, čo znamená, že celková dĺžka hovoru bola 8 sekúnd.

Tabuľka 2: Príklady niektorých CDR záznamov

Atribút	Príklad	Význam
accountcode	12345	ID účtu
src	2101	Číslo volajúceho účastníka.
billsec	60	Počet sekúnd od zdvihnutia až po koniec hovoru.
amaflag	DOCUMENTATION	Hovory možno označiť parametrami DOCUMENTATION, BILLING, IGNORE.
channel	SIP/0004F2040808-a1bc23ef	Informácie o použítom prichádzajúcom kanále.
clid	"Ladislav Behán" <2222>	Telefónne číslo + meno užívateľa.
dcontext	SPAIN	Použitý kontext z extensions.conf.
disposition	ANSWERED	informácia čo sa stalo s hovorom: ANSWERED, NO ANSWER, BUSY, FAILED.
dst	2102	Číslo volaného účastníka.
dstchannel	SIP/2101-0000000f	obsahuje informácie o použítom odchádzajúcom kanále.
duration	195	celkový využitý čas Asterisku začatý od vytočenia volaného do doby ukončenia hovoru.
lastapp	Dial	Posledný príkaz asterisku pred ukončením hovoru.
lastdata	SIP/0004F2046969,30,tT	Posledné argumenty aplikácie.
start	2016-03-13 12:00:00	Začiatok hovoru.
answer	2016-03-13 12:00:10	Data o zdvihnutí hovoru volaným účastníkom.
end	2016-03-13 12:02:16	Koniec hovoru.
uniqueid	1288112400.1	Asterisk je schopný po úprave generovať jedinečné identifikácie hovoru.

Pre spracovanie týchto záznamov som vytvoril C++ aplikáciu `cdr_manager.cpp`, ktorá tieto CDR záznamy parsuje a ukladá do rôznych databázi, ktoré sú využívané v samotnej realizácii vlastného účtovacieho systému.

Akonáhle asterisk uloží CDR záznam o hovore do súboru `Master.csv` program `cdr_manager.cpp` si v tom okamžiku načíta posledný riadok z tohto súboru. V prvom rade odstráni z reťazca všetky čiarky, ktoré oddeľujú od seba jednotlivé záznamy, s ktorými ďalej bude manipulovať. V ďalšej časti zistí celkovú dĺžku hovoru, ktorú získa výpočtom rozdielu medzi časom ukončenia hovoru a časom kedy bol hovor akceptovaný volanou stranou. Následne si program vyžiada informácie z databáze `services.db` o užívateľovi, ktorý hovor inicioval. Jedná sa o výšku kreditu poprípade o stave predplatených minút. Pokiaľ užívateľ využíva službu predplateného kreditu je mu vypočítaná celková suma za hovor a tá je následne odčítaná od pôvodnej sumy. Užívateľovi disponujúcemu predplateným paušálom je celkový čas hovoru odčítaný od zvyšku jeho voľných minút. Výška výslednej sumy je závislá od toho, či bol hovor realizovaný v sieti OpenBTS alebo telefón v sieti OpenBTS volal na IP telefón alebo poslednou možnosťou je či bol uskutočnený cez SIP/IAX trunk. Jednotlivé ceny hovorov pri rôznych scenároch je možné vidieť v tabuľke 3.

Tabuľka 3: *Ceny hovoru pri jednotlivých scenároch použitia*

Typ hovoru	Cena pre kreditového zákazníka	Cena pre paušalného zákazníka
v sieti OpenBTS	0,5 Kč	prevolaný čas * 1
na IP telefón	0,75 Kč	prevolaný čas * 1
cez SIP trunk	1 Kč	prevolaný čas * 1
cez IAX trunk	2 Kč	prevolaný čas * 2

Databáza s informáciami o predplatených službách je nakoniec aktualizovaná o nový stav účtu a v záverečnej fáze program prostredníctvom funkcie `mysqlcallingData()` skontroluje či databáza užívateľa v tvare `USER_data.db` už bola predtým vytvorená. V prípade, že databáza nebola nájdená program ju vytvorí a následne do tejto novo vytvorenej MySQL databáze vloží všetky spracované data z CDR záznamu, aby si ich neskôr mohol zákazník pozrieť alebo stiahnuť súbor PDF obsahujúci jeho fakturačné údaje.

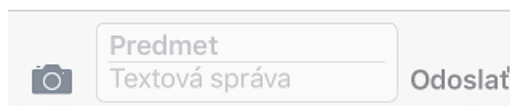
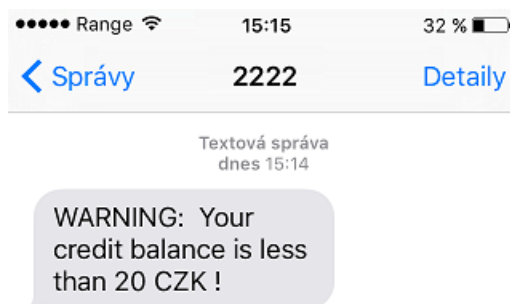
Tabuľka 4: *Štruktúra MySQL databáze USER_data.db*

Názov atribútu	Dátový typ
ID	int(11)
User	varchar(255)
Country	varchar(255)
Called	varchar(255)
Start	varchar(255)
End	varchar(255)
Duration	varchar(255)
Price	varchar(255)

Atribút ID predstavuje automaticky generované číslo záznamu, User atribút obsahuje užívateľove telefónne číslo v sieti OpenBTS, položkou Country môžeme chápať ako krajinu kde bol hovor smerovaný, ako príklad som si v tejto diplomovej práci som si zvolil, že asterisk server v Českej Republike je spojený prostredníctvom SIP trunku so španielskym asterisk serverom a so slovenským serverom ho spája SIP trunk. Ďalej Called je atribút obsahujúci číslo volaného účastníka, Start a End sú atribúty pre začiatok a koniec hovoru, Duration predstavuje celkovú dĺžku hovoru a záverečný atribút tejto databáze je Price, do ktorého je ukládaná celková suma uskutočneného hovoru.

```
SUCCESS: SQLITE DB was successfully OPENED!  
SUCCESS: Username found in DB!  
INFO : The call lasted : 68 seconds  
INFO : User -> 2101 has service -> cash  
INFO : Updating credit status...  
INFO : Call was made in Czech Republic, prize -> 34 CZK  
INFO : Updating users table...  
SUCCESS: SQLITE DB was successfully UPDATED!  
SUCCESS: MYSQL DB was successfully OPENED!  
SUCCESS: MYSQL DB was successfully UPDATED!  
SUCCESS: MYSQL DB was successfully CLOSED!  
SUCCESS: MYSQL DB was successfully OPENED!  
SUCCESS: MYSQL DB call data successfully INSERTED!  
SUCCESS: MYSQL DB number of calls was UPDATED!  
SUCCESS: SQLITE DB was successfully CLOSED!
```

Obrázok 14: Priebeh programu *cdr_manager.cpp*



Obrázok 15: Notifikačná textová správa o nízkom stave kreditu

4.6 Kontrola prekročenia predplatených služieb pri hovore

Ďalšiu dôležitou súčasťou tohto účtovacieho systému je kontrola, či pri realizácii telefónneho hovoru volaným účastníkom náhodou vzhľadom na jeho predplatené služby nedošlo k ich

prekročení. Pre túto funkciu som implementoval program `call_manager.cpp`, ktorá neustále kontroluje obsadenosť hovorových kanálov systému asterisk.

Ako som spomínal, tak základným kameňom celého procesu je kontrola, či došlo k zriadeniu hovorového kanálu. Stav aktívnych kanálov môžem kontrolovať prostredníctvom príkazu `core show channel` zadaného do príkazového riadku asterisku. Zadaním nasledujúceho terminálového príkazu zaistím pravidelné preverovanie všetkých zriadených kanálov smerovaných cez systém asterisk. Výpisy z terminálu následne ukládam do súboru `active_channels.txt`, ktorý je neustále kontrolovaný spomínaným programom `call_manager.cpp`.

```
watch -n 1 "sudo asterisk rx 'core show channels' > active_channels.txt"
```

Výpis z asterisk CLI, ktorý je uložený do súboru `active_channels.txt` vypadá nasledovne:

```
Channel                Location                State  Application(Data)
SIP/2101-0000000f      3333@SPAIN:2           Up     Dial(SIP/333,30)
SIP/3333-00000010      (None)                  Up     AppDial((Outgoing
Line))
2 active channels
```

Celý proces začína opakovanou kontrolou spomínaného textového súboru. Program neustále preveruje, či sa v súbore neobjaví aktívny hovorový kanál. V prípade jeho nájdenia, môžeme vidieť, že z výpisu môžeme zistiť množstvo dôležitých údajov pre spracovanie. Celý priebeh programu je možné vidieť na obrázku 16.

V prvom rade si program načíta telefónne číslo volajúceho účastníka, ktoré je následne porovnané s `sqlite3` databázou `services.db`, či sa volajúci nachádza v tejto databáze. V prípade, že program daného účastníka nenájde, program hovorový kanál okamžite ukončí. V prípade zhody so záznamom uloženým v databáze si program načíta akou predplatenou službou volajúci účastník disponuje, či má zaplatený kredit alebo paušál s predplatenými minútami.

```
SUCCESS: Database was successfully opened!
SUCCESS: Active channel found!!
INFO    : Active channel is used by -> 2101 <=> 2102
INFO    : 2101 has service -> cash
INFO    : User 2101 is allowed to use this channel for 60 secs
INFO    : Allowed time was exceeded -> terminating active channel!!!
INFO    : Text message was sent to user!
INFO    : Updating credit status for user: 2101
SUCCESS: Updating database..OK!
INFO    : User -> 2101 don't have enough money to make a call!!!
```

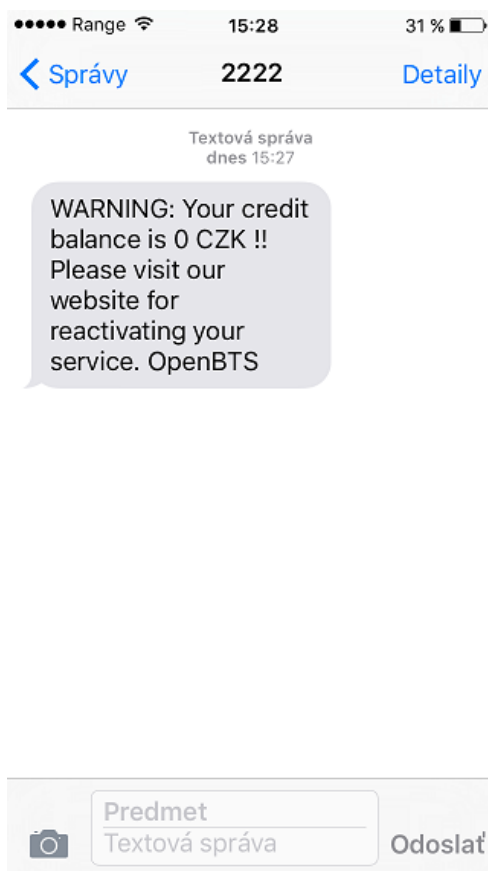
Obrázok 16: Priebeh programu `call_manager.cpp`

Ďalej si program zistí, či je hovor smerovaný cez SIP/IAX trunk, na IP telefón alebo je uskutočnený v sieti OpenBTS. Na základe týchto zistených údajov je účastníkovi určená doba, po ktorú môže byť tento hovorový kanál aktívny. Program ďalej neustále kontroluje, či nedošlo

k ukončeniu hovoru pred vyčerpaním limitu predplatených služieb. Ak došlo k prečerpaniu časového limitu, aktívny kanál je okamžite zrušený, program si opäť otvorí databázu obsahujúcu údaje o účastníkových predplatených službách a nastaví ich na nulovú hodnotu.

```
string kill_channel = "screen -S asterisk -X stuff ' channel request hangup SIP/2101-0000000b \n'";  
  
system(kill_channel.c_str());  
  
string sms = "screen -S openbts -X stuff ' sendsms 2222 230024701231666  
`WARNING: Your credit balance is 0 CZK !! Please visit our website for  
reactivating your service. OpenBTS`\n'";  
  
system(sms.c_str());
```

Predošlé riadky zdrojového kódu programu poukazujú ako sa daný kanál korektne ukončuje a ako je zákazník v tom okamžiku prostredníctvom SMS správy informovaný o nulovom stave jeho účtu, ktorou je taktiež vyzvaný k návšteve webovej stránky operátora z dôvodu obnovy svojej predplatenej služby, pretože inak hlasové služby operátora nebude schopný naďalej používať.



Obrázok 17: Notifikačná sms o nulovom stave kreditu

Ďalej je potrebné ešte zaistiť, aby pri ďalšiom vytočení čísla tento účastník sa nemohol dovolať žiadnemu účastníkovi dokým neobnoví svoje hlasové služby. Zabezpečím to úpravou konfiguračného súboru `extensions.conf` podľa nasledujúcich riadkov. Nasledujúcou úpravou zabezpečím, že pri vytočení ktoréhokoľvek čísla je najskôr overené či volaný účastník náhodou nedisponuje kreditom s nulovou hodnotou. V prípade, že jeho kredit je nulový hovor je ukončený, v opačnom prípade Asterisk pokračuje nasledujúcim príkazom `Dial`.

```
[sip-external]
exten => N.,1,NoOp()
exten => 2101,n,Set(kred=${ODBC_SQL(select cash from kredit where username
= \"${EXTEN}\")} )
exten => N.,n,GotoIf("${kred}" = "0"?zaves)
exten => N.,n,Dial(SIP/${EXTEN},30)
exten => N.,n,Congestion()
exten => N.,n(zaves),Hangup()
```

4.7 Spôsob kontroly odoslaných textových správ

Ďalším problémom, ktoré bolo potrebné vyriešiť pri realizácii tohto účtovacieho systému bolo, akým spôsobom budem monitorovať odoslané textové správy. V tomto prípade využívam súbor `OpenBTS.log` nachádzajúci v adresári `/var/log/`. Jedná sa o logovací súbor, do ktorého OpenBTS ukladá všetky svoje logovacie správy. Nasledujúcim príkazom pravidelne sledujem tento súbor a vyberám si z neho len riadky s potrebným obsahom, ktorý mi prezradza, že bola užívateľom odoslaná textová správa.

```
watch -n 1 "sudo grep "Got SMS rqst qtag" /var/log/OpenBTS.log >
sms_result.txt"
```

Ako je možné si z príkazu všimnúť, riadky s potrebným obsahom ukladám do textového súboru `sms_result.txt`. Tento textový súbor je následne spracovávaný programom `sms_manager.cpp`. Program tento súbor neustále kontroluje a zisťuje, či v sieti OpenBTS niektorý účastník odoslal textovú správu. Riadok prezradzajúci poslanie správy vyzerá nasledovne:

```
Mar  2 15:03:22 studentPC smqueue: NOTICE 4842:4848 2016-03-02T15:03:22.1
smqueue.cpp:2455:main_loop: Got SMS rqst qtag '658054--
OBTSbwgdfkphscqborwe' from IMSI230024701231026 $
```

Program `sms_manager.cpp` si v počiatočnej fáze načíta vždy posledný riadok textového súboru. Následne si tento riadok spracuje a vybere si z neho dve podstatné informácie. Najdôležitejšiou informáciou je IMSI číslo užívateľa. Program na základe tohto čísla skontroluje tabuľku `users` v databáze `sqlite3.db`, či sa IMSI číslo v databáze nachádza. Pokiaľ bolo nájdené načíta sa pridelené telefónne číslo, predplatená služba a jej stav. Pokiaľ má užívateľ predplatenú paušálnu

službu, tak je načítaný stav jeho zostávajúcich voľných sms správ a tento stav je následne znížený o 1. V prípade, že užívateľ má zaplatený kredit, tak sú mu zo zostavajúcej čiastky odrátané 3 koruny. Program nakoniec ešte aktualizuje databázu, ktorá poukazuje na počet textových správ odoslaných užívateľom siete OpenBTS za mesačné obdobie. Tieto štatistiky sú mu potom v grafickej podobe dostupné na webových stránkach operátora.

```
INFO    : User with imsi number IMSI230024701231026 sent sms!  
SUCCESS: SQLITE DB was successfully OPENED!  
INFO    : His phone number is -> 2101  
INFO    : Updating Sqlite table...  
SUCCESS: SQLITE DB was successfully UPDATED!  
INFO    : Updating MySQL table...  
SUCCESS: MYSQL DB number of text messages was successfully UPDATED!
```

Obrázok 18: Priebeh programu *sms_manager.cpp*

5 Riadenie prístupu k službám na základe účtovania

V tejto kapitole predstavím ako môže užívateľ spravovať svoje konto, čiže aktivovať rôzne služby a ich následne zaplatenie alebo si môže pozrieť svoje štatistiky či už graficky alebo textovo stiahnutím faktúry vo forme PDF súboru.



Obrázok 19: Logo operátora

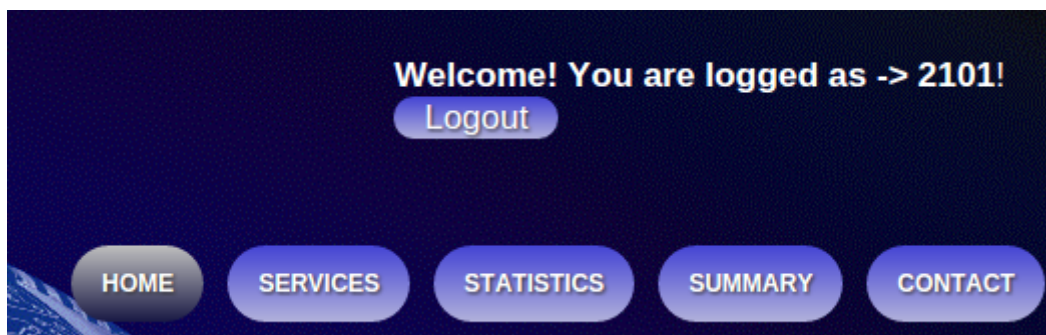
Ako som spomínal v predchádzajúcej kapitole, tak každý účastník siete OpenBTS pri prihlásení do tejto siete automaticky obdrží uvítaciu SMS textovú správu obsahujúcu jeho automaticky vygenerované telefónne číslo a zároveň IMSI číslo jeho SIM karty. Na základe týchto údajov sa na webových stránkach operátora prihlási ku svojmu účtu. Ako je vidno na obrázku 20, užívateľ pri otvorení stránky operátora je požiadaný o zadanie práve týchto dvoch údajov, aby mohol prístupit' ku aktivácii svojich predplatených služieb.

Obrázok 20: Prihlasovacia stránka OpenBTS

Ako je vidno na obrázku 21 tak po úspešnom prihlásení je zobrazený uvítací odkaz. Kliknutím na tento obrázok sa užívateľ dostane na hlavnú webovú stránku tohto operátora. Na hlavnej stránke je užívateľovi zobrazená ponuka so záložkami Home, Services, Statistics, Summary a na záver Contact.

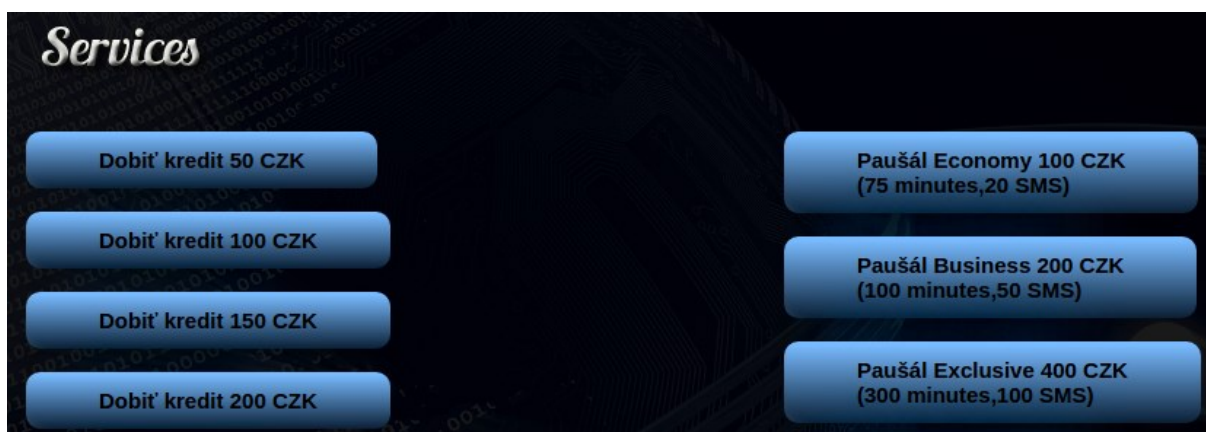


Obrázok 21: *Uvítací odkaz na webových stránkach operátora*



Obrázok 22: *Hlavné menu*

Po otvorení záložky Services sa užívateľovi zobrazí ponuka na aktiváciu rôznych hlasových služieb. Môže si zvoliť buď zo štyroch typov kreditných služieb alebo si môže zaplatiť niektorú z paušálnych prepustených hlasových služieb. Každé jedno tlačidlo obsahujúce predplatenú hlasovú službu je prepojené na internetový platobný systém. To znamená, že zákazník po zvolení svojej preferovanej služby bude presmerovaný na webovú stránku tejto finančnej internetovej služby, kde bude vyzvaný k uhradeniu sumy za požadovaný tarif, viz obrázok 24.



Obrázok 23: *Aktivácia predplatených hlasových služieb*

Your Shopping Cart

Description	Item price	Quantity	Amount
Kredit 100 Item # 100	100.00	1 Update Remove	100.00
Item total:			100.00


[Continue Shopping](#) [Check Out](#) OR [Check out with PayPal](#)
The safer, easier way to pay

Obrázok 24: Platba vybranej služby prostredníctvom PayPal

V záložke Summary je možné si spätne prezrieť všetky uskutočnené hovory. V tejto tabuľke je detailne popísané komu užívateľ volal a kde bol tento hovor smerovaný. Následne si môže prezrieť celkové trvanie hovoru s upresneným začiatkom a koncom hovoru. Posledná položka Price poukazuje na účtovanú sumu za uskutočnený telefonát. Pod tabuľkou je umiestnený odkaz s nápisom Download PDF. Kliknutím na tento odkaz sa spustí sťahovanie celého prehľadu vo formáte PDF. Tieto súbory sú vytvárané prostredníctvom nástroja FPDF spomínaného v predchádzajúcej kapitole.

Summary

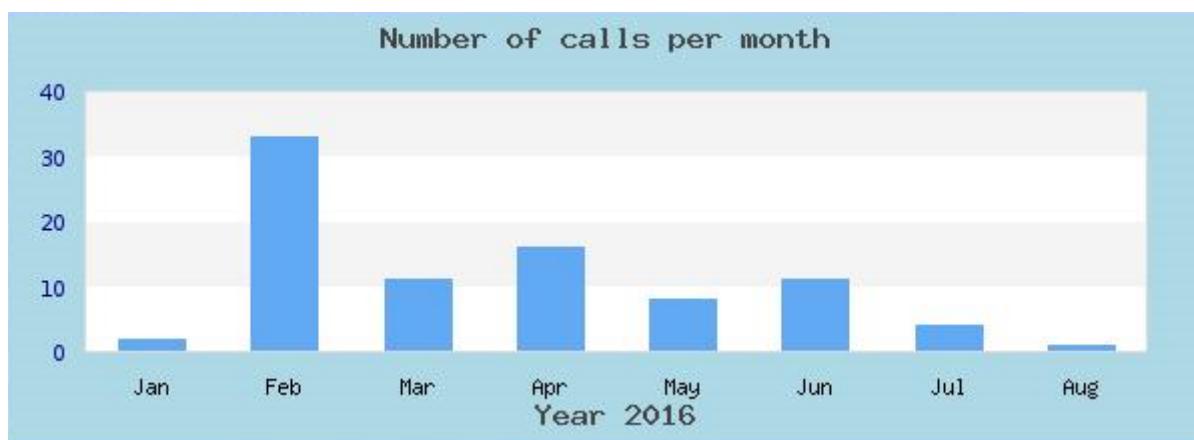
Id	User	Country	Called	Start	End	Duration	Price
1.	2101	CzechRep	2102	16:51:14	16:52:22	8	4
2.	2101	CzechRep	2102	16:53:28	16:54:20	52	26
3.	2101	SPAIN	3333	17:00:18	17:00:56	38	38
4.	2101	SPAIN	3333	17:02:16	17:02:48	32	16
5.	2101	SPAIN	3333	17:05:01	17:06:23	82	41
6.	2101	IPPHONE	5555	14:25:31	14:26:17	46	35
7.	2101	IPPHONE	5555	18:41:12	18:43:11	59	44
8.	2101	SLOVAKIA	4444	20:21:33	20:22:49	76	152

[Download PDF](#) 

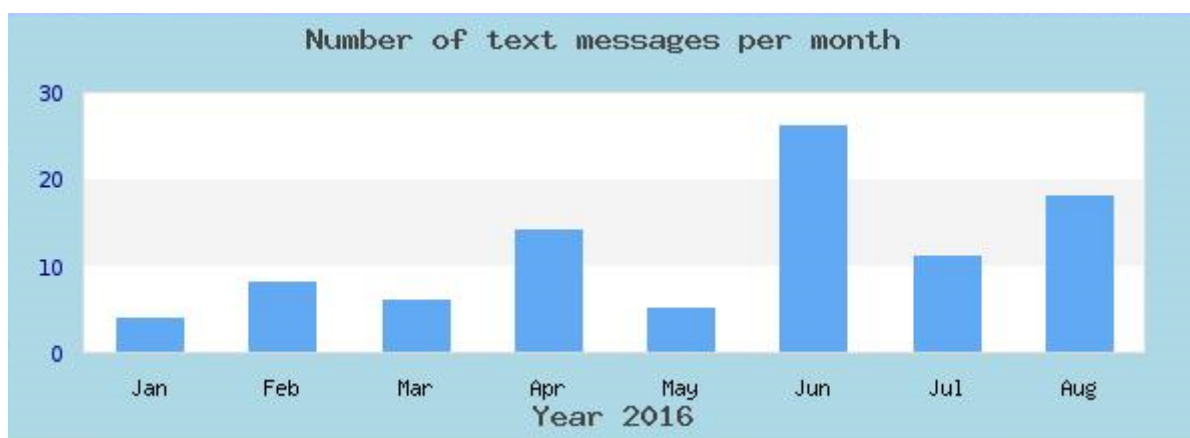
Obrázok 25: Prehľad všetkých uskutočnených hovorov

Predposledná záložka je tvorená jednotlivými štatistikami. Zákazník si môže napríklad prezrieť počet hovorov uskutočnených v jednotlivých mesiacoch, taktiež počet textových správ a celkový počet finančných nákladov, ktoré investoval do svojich predplatených hlasových služieb.

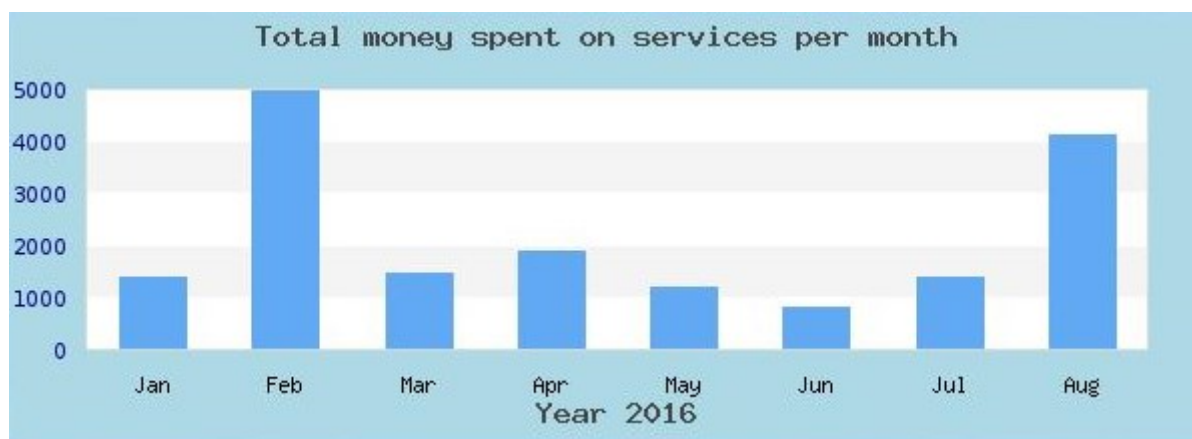
Jednotlivé grafy generované prostredníctvom objektovo-orientovanej knižnice PHP Jpgraph, ktorá pre ich spracovanie využíva data načítane z SQL databáze.



Obrázok 26: Počet hovorov v jednotlivých mesiacoch



Obrázok 27: Počet odoslaných textových správ v jednotlivých mesiacoch



Obrázok 28: Celkové množstvo investovaných peňazí do hlasových služieb

6 Záver a zhodnotenie dosiahnutých výsledkov

Vo svojej diplomovej práci som sa venoval téme predplatených hlasových služieb na platforme openBTS. V úvode tejto práce som sa venoval teoretickému úvodu do tejto softwarovo riešenej GSM siete. Popísal som všetky potrebné komponenty, ktoré sú nutné pre realizáciu tejto siete, ktorej hlavným stavebným kameňom je USRP N210. Inštalácia systému bola oproti mojej bakalárskej práci trochu pozmenená a je taktiež uvedená vo štvrtej kapitole tejto práce.

V následovnej kapitole som sa venoval open-source účtovacím systémom, každý som jednotlivito popísal a predstavil výhody a nevýhody. Svoju pozornosť som hlavne zameril na účtovací systém A2Billing, ktorý sa radí v dnešnej dobe medzi najpoužívanejšie a najznámejšie účtovacie systémy. A2Billing je skutočne prepracovaný systém poskytujúci množstvo funkcií, na ktorom oceňujem hlavne prehľadné grafické webové rozhranie. Medzi jeho nevýhody však patrí skutočnosť, že niektoré funkcie sú platené a taktiež prípadná podpora alebo asistencia pri implementácii pre poskytovateľa nie je zadarmo.

Hlavným cieľom mojej diplomovej práce bolo však vytvorenie vlastného účtovacieho systému, ktorý bude korektne fungovať v spolupráci s platformou OpenBTS. V samotnej praktickej realizácii som najskôr vylepšil registráciu účastníkov tejto siete. V bakalárskej práci som implementoval program, ktorý umožňoval manuálnu registráciu. Tento program na základe IMSI čísla ako vstupného parametru, upravil všetky potrebné konfiguračné súbory Asterisku a zároveň aj jeho databázu. Následne si už mohli užívatelia v sieti OpenBTS navzájom telefonovať. V diplomovej práci som tento proces plne automatizoval. Zistil som, že pri registrácii telefónu do tejto siete sa jeho IMSI číslo automaticky ukladá do databáza TMSItable.db. Databázu neustále kontrolujem programom openbts_reg.cpp, ktorý pri zistení nového registrovaného IMSI čísla už potom podobne ako v predchádzajúcom prípade upraví všetky potrebné konfiguračné súbory a databáze.

V ďalšom bode diplomovej práce som sa venoval otázke, akým spôsobom bude prebiehať účtovanie hovorného v tejto sieti. OpenBTS však pre svoju funkčnosť a smerovanie hovorov využíva pobočkovú ústredňu Asterisk. Tento nástroj po skončení každého hovoru ukladá informácie o jeho priebehu vo forme CDR záznamov do CSV súboru. CDR záznamy následne spracovávam programom cdr_manager.cpp, ktorý tieto spracované údaje následne uloží do rôznych databázi, s ktorými je potom v rámci účtovacieho systému rôzne manipulované.

Naskytá sa však otázka, ako by som tieto CDR data zbieral a triedil v prípade, že by takto vytvorená sieť pozostávala z viacerých BTS staníc. Pokiaľ by každá BTS stanica bola pripojená k rozdielnému Asterisk serveru, tak by podobne ako v tomto prípade boli data spracovávané a ukládané do MySQL databáze vlastnej pre jednotlivé servery a následne by bol implementovaný

master program, ktorý by data z jednotlivých serverov sťahoval, spracovával a triedil na základe prideleného identifikačného čísla.

Ďalšia časť praktickej realizácie je venovaná implementácii programu, ktorý bude kontrolovať či užívateľ náhodou pri telefonovaní neprekročil limit svojich predplatených služieb. Program `call_manager.cpp` na základe stavu konta užívateľa vypočíta po akú dobu môže volajúci využívať hovorový kanál. V prípade prekročenia limitu dochádza k vynútenému ukončeniu spojenia a volajúci je informovaný o nulovom stave svojho kreditu. Zákazník je dokonca aj vopred upozornený na nízky stav kreditu, pokiaľ jeho zostávajúca čiastka je rovná alebo menšia ako 20 korún.

Ďalším C++ programom, ktorý som implementoval pri realizácii tejto diplomovej práce je `sms_manager.cpp`. Tento program neustále kontroluje logovací súbor `OpenBTS.log`, ktorý obsahuje potrebné informácie o odoslaných textových správach. Na základe IMSI čísla je potom užívateľovi odčítaný poplatok za textovú správu alebo v prípade využívania paušálnej služby, je od jeho stavu počtu voľných textových správ odčítaná jedna správa.

Posledná kapitola diplomovej práce je venovaná riadeniu prístupu k službám na základe účtovania. Pri registrácii užívateľa do siete OpenBTS mu je následne odoslaná textová správa s jeho automaticky vygenerovaným telefónnym číslom a IMSI číslom jeho SIM karty. Využitím týchto údajov sa môže prihlásiť ku svojmu účtu, ktorý bol automaticky vygenerovaný na webových stránkach operátora. Následne si môže svoj účet aktivovať o predplatený kredit alebo môže využiť ponuku niektorého z pripravených paušálov. Ďalej si môže prezrieť informácie o všetkých realizovaných telefonátoch a taktiež si celý tento prehľad môže stiahnuť vo formáte PDF do svojho počítača. Účet užívateľa ešte obsahuje rôzne graficky spracované štatistiky o počte hovorov, textových správ a celkových finančných nákladov za jednotlivé mesiace.

V tejto diplomovej práci som využil širokú škálu či už programovacích jazykov ako napr. C++, PHP, HTML, CSS, ale aj rôznych nástrojov, ktoré sú všetky podrobne popísané v jednotlivých kapitolách tejto práce. Výsledkom je funkčný nízkonákladový mobilný operátor, ku ktorého úplnosti snáď len chýba integrácia dátových služieb, ktorú by bolo možné realizovať interoperabilitou systému OpenBTS s prvkami OsmoSGSN a OpenGGSN z projektu Osmocom, o ktorom je možné sa dočítať viac v mojej bakalárskej práci. Na záver by som už len dodal, že túto diplomovú prácu som si sám navrhol a všetky ciele, ktoré som si na začiatku jej realizácie stanovil, som do poslednej bodky splnil.

7 Použitá literatura

- [1] BEHÁŇ, Ladislav. *OpenBTS a její integrace do telekomunikační sítě*. Ostrava, 2014. Bakalářská práce. VŠB - Technická univerzita Ostrava. Vedoucí práce Doc. Ing. Miroslav Vozňák, Ph.D.
- [2] BURGESS, D. A, SAMRA H. The Open BTS Project – an opensource GSM base station. September 2008. [Online]. Dostupné z: <http://www.ahzf.de/itstuff/papers/OpenBTSProject.pdf>
- [3] OpenBTS. [Online]. Dostupné z: <http://gnuradio.org/redmine/projects/gnuradio/wiki/OpenBTSBackground>
- [4] Range Networks. [Online]. Dostupné z: <http://www.rangenetworks.com>
- [5] Meggelen, J.V, Smith, J., Madsen, L. Asterisk™ The Future of Telephony. Sevastopol: O'Reilly Media, Inc., 2005. ISBN 0-596-00962-3
- [6] KEMETMULLER Ch., SEEGER M., BAIER H., BUSCH Ch. Manipulating Mobile Devices with a private GSM Base Station - a Case Study. In Proceedings 8th International Network Conference (INC 2010), 2010.
- [7] M. Vozňák, "Technologie a protokoly multimediálních komunikací pro integrovanou výuku VUT a VŠB-TUO." VŠB-TU Ostrava, 2014, 252 str., ISBN 978-80-248-3326-2.
- [8] KOSEK, Jiří. PHP :Tvorba interaktivních internetových aplikací. Podrobný průvodce /Praha :Grada Publishing,1998. 1. vyd. 490 s. ISBN 8071693731
- [9] A2Billing Install Guide. [Online]. Dostupné z: <https://www.powerpbx.org/content/a2billing-install-guide-v27>
- [10] Installing and Configuring PHP JpGraph on Linux. [Online]. Dostupné z: <http://queirozf.com/entries/installing-and-configuring-php-jpgraph-on-linux>
- [11] Installing Jpgraph [Online]. Dostupné z: <http://jpgraph.net/download/manuals/chunkhtml/ch03s03.html>
- [12] Installing Apache2 With PHP5 And MySQL Support On Ubuntu 14.04LTS (LAMP). [Online]. Dostupné z: <https://www.howtoforge.com/ubuntu-lamp-server-with-apache2-php5-mysql-on-14.04-lts>
- [13] ČERNÝ, Jan. *Systém účtování hovorného pro open-source IP PBX*. Ostrava, 2011. Diplomová Práce. VŠB - Technická univerzita Ostrava. Vedoucí práce Doc. Ing. Miroslav Vozňák, Ph.D.
- [14] DEPIAK, Petr. *BILINGOVÝ SYSTÉM A MONITOROVÁNÍ HOVORŮ PRO PBX ASTERISK*. Brno, 2010. Diplomová Práce. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ. Vedoucí práce Ing. ONDŘEJ KRAJSA.

-
- [15] STOKLAS, Martin. *Podrobné záznamy o volání a jejich statistické zpracování*. Ostrava, 2012. Bakalářská práce. VŠB – Technická univerzita Ostrava. Vedoucí práce Doc. Ing. Miroslav Vozňák, Ph.D.
- [16] RABIENCNÝ, Tomáš. *Problematika billingu v IP telefonii*. Praha, 2010. Bakalářská práce. Bankovní institut vysoká škola Praha. Vedoucí práce Ing. Vladimír Beneš.
- [17] Open Source Billing Systems Dostupné z: <http://www.voip-info.org/wiki/view/Open+Source+Billing+Systems>
- [18] PULGARÍN, J.E., PARRA, E., QUINTERO, A.G. *GSM system implementation for academic purposes (2013)*. 2013 IEEE Colombian Conference on Communications and Computing, COLCOM 2013 - Conference Proceedings, art. no. 6564840, .DOI: 10.1109/ColComCon.2013.6564840
- [19] LI, Z., TANG, J., ZHU, X., KAI, C. *Simple GSM base station based on universal software radio peripheral (2014)*. 5th International Conference on Computing Communication and Networking Technologies, ICCCNT 2014, art. no. 6963047. DOI: 10.1109/ICCCNT.2014.6963047
- [20] FAJKUS, M., MIKULEC, M., VOZNAK, M., TOMIS, M., FAZIO, P. *Speech quality measurement of GSM infrastructure built on USRP N210 and openBTS project (2014)*. Advances in Electrical and Electronic Engineering, 12 (4), pp. 341-346. DOI: 10.15598/aeec.v4i4.1215
- [21] PARTILA, P., KOHUT, M., VOZNAK, M., MIKULEC, M., SAFARIK, J., TOMALA, K. *A methodology for measuring voice quality using PESQ and interactive voice response in the GSM channel designed by openBTS (2013)*. Advances in Electrical and Electronic Engineering, 11 (5), pp. 380-386. DOI: 10.15598/aeec.v11i5.894

8 Zoznam príloh

Príloha A:	faktúra vo forme stiahnuteľného PDF súboru.....	L
Príloha B:	Alternatívna inštalácia OpenBTS.....	LI
Príloha C:	Asterisk konfiguračný súbor sip.conf.....	LIII
Príloha D:	Asterisk konfiguračný súbor extensions.conf.....	LV
Príloha E:	Asterisk konfiguračný súbor iax.conf.....	LVII

Súčasťou diplomovej práce je CD.

Adresárová štruktúra priloženého CD:

- Registrácia/openbts_reg.cpp
- Registrácia/sqlite3.h
- Registrácia/sqlite3.c
- Registrácia/Makefile
- Registrácia/sqlite3.db
- Registrácia/sms_manager.cpp
- Registrácia/call_manager.cpp
- Registrácia/cdr_manager.cpp
- www/html/index.php
- www/html/summary_member.php
- www/html/statistics_member.php
- www/html/graf1.php
- www/html/graf2.php
- www/html/graf3.php
- www/html/services_member.php
- www/html/pdf.php
- www/html/pausa11.php
- www/html/pausa12.php
- www/html/pausa13.php
- www/html/member.php
- www/html/logout.php
- www/html/login.php
- www/html/kredit.php
- www/html/contact_member.php
- DP/BEH032.pdf

Príloha A: faktúra vo forme stiahnuteľného PDF súboru



SUMMARY OF CALLS

Id	User	Country	Called	Start	End	Duration (sec)	Price(CZK)
1	2101	CzechRep	2102	16:51:14	16:52:22	8	4
2	2101	CzechRep	2102	16:53:28	16:54:20	52	26
3	2101	SPAIN	3333	17:00:18	17:00:56	38	38
4	2101	SPAIN	3333	17:02:16	17:02:48	32	16
5	2101	SPAIN	3333	17:05:01	17:06:23	82	41
6	2101	IPPHONE	5555	14:25:31	14:26:17	46	35
7	2101	IPPHONE	5555	18:41:12	18:43:11	59	44
8	2101	SLOVAKIA	4444	20:21:33	20:22:49	76	152

Príloha B: Alternatívna inštalácia OpenBTS

Inštalácia potrebných závislostí:

```
$ sudo apt-get install software-properties-common python-software-properties
$ sudo add-apt-repository ppa:git-core/ppa
```

Aktualizácia systému:

```
$ sudo apt-get update
```

Inštalácia GIT:

```
$ sudo apt-get install git
```

```
git clone https://github.com/RangeNetworks/dev.git
```

Overenie verzie GIT:

```
$ git --version
git version 1.9.1
$ ./switchto.sh master
```

Spustenie skriptu build.sh pre konkrétny typ USRP:

```
./build.sh N210
```

Extrahovanie súboru:

```
sudo dpkg -i BUILD-timestamp/*.deb
```

Inštalácia:

```
sudo apt-get -f install
```

Štart jednotlivých nainštalovaných komponent OpenBTS:

```
$ sudo start sipauthserve
$ sudo start smqueue
$ sudo start openbts
$ sudo start asterisk
```

Pozastavenie aplikácii:

```
$ sudo stop sipauthserve
```

```
$ sudo stop smqueue
```

```
$ sudo stop openbts
```

```
$ sudo stop asterisk
```

Príloha C: Asterisk konfiguračný súbor sip.conf

```
[general]
port=5060
bandwidth=low
jitterbuffer=yes

tlsenable=yes                ;povolenie TLS
tlsbindaddr=0.0.0.0         ;pripojenie TLS k IP
tlscertfile=/etc/asterisk/keys/SPAIN.pem ;cesta k certifikátu
tlsdontverifyserver=yes ;overenie či sa Asterisk správa jako klient
tlscipher=DES-CBC3-SHA ;šifrovací algoritmus
tlsclientmethod=tlsv1      ; verzia TLS/SSL

register=>tls://SPAIN:trunk@158.196.244.143:5061

tos_sip=cs3
tos_audio=ef
tos_video=af41
tos_text=af41
cos_sip=3
cos_audio=5
cos_video=4
cos_text=3
maxexpiry=3600
minexpiry=60
defaultexpiry=3600
dynamic_exclude_static=yes
use_q850_reason=yes
rtptimeout=60
rtpholdtimeout=300
allowoverlap=no

disallow=all
allow=gsm
allow=alaw
```

allow=g711
relaxdtmf=yes
dtmfmode=auto

[SPAIN]
type=friend
host=dynamic
trunk=yes
username=SPAIN
secret=trunk
qualify=yes
context=internal
dtmfmode=rfc2833

[IMSI230056332177132]
callerid=2102
type=friend
qualify=yes
host=dynamic
context=CzechRep
username= IMSI230056332177132

[IMSI230026715846389]
callerid=2101
type=friend
qualify=yes
host=dynamic
context=CzechRep
username= IMSI230026715846389

Príloha D: Asterisk konfiguračný súbor extensions.conf

```
[globals]

[general]
autofallthrough=yes
[default]

[sip]
include => internal
include => remote

[outbound-trunk]
exten => _N.,1,Answer()

[internal]
exten => _N.,1,Set(Name=${ODBC_SQL(select dial from dialdata_table where
exten = \"${EXTEN}\")})
exten => _N.,n,GotoIf("${Name}" = "") ?outbound-trunk,${EXTEN},1)
exten => _N.,n,Set(IPAddr=${ODBC_SQL(select ipaddr from sip_buddies where
name = \"${Name}\")})
exten => _N.,n,GotoIf("${IPAddr}" = "") ?outbound-trunk,${EXTEN},1)
exten => _N.,n,Dial(SIP/${Name}@${IPAddr}:5062)

[CzechRep]
exten => 2101,1,NoOp()
exten => 2101,n,Set(kred=${ODBC_SQL(select cash from kredit where username
= \"${EXTEN}\")})
exten => 2101,n,GotoIf("${kred}" = "0"?zaves)
exten => 2101,n,Dial(SIP/ IMSI230026700507478@127.0.0.1:5062)
exten => 2101,n,Congestion()
exten => 2101,n(zaves),Hangup()

exten => 2102,1,NoOp()
exten => 2102,n,Set(kred=${ODBC_SQL(select cash from kredit where username
= \"${EXTEN}\")})
exten => 2102,n,GotoIf("${kred}" = "0"?zaves)
exten => 2102,n,Dial(SIP/IMSI230025900646538@127.0.0.1:5062)
exten => 2102,n,Congestion()
```

exten => 2102,n(zaves),Hangup()

[SLOVAKIA]

exten => 4444,1,NoOp()

exten => 4444,n,Set(kred=\${ODBC_SQL(select cash from kredit where username = \"\${EXTEN}\")})

exten => 4444,n,GotoIf("\${kred} = \"0\")?zaves)

exten => 4444,n,Dial(IAX2/SLOVAKIA/\${EXTEN})

exten => 4444,n,Congestion()

exten => 4444,n(zaves),Hangup()

[SPAIN]

exten => 3333,1,NoOp()

exten => 3333,n,Set(kred=\${ODBC_SQL(select cash from kredit where username = \"\${EXTEN}\")})

exten => 3333,n,GotoIf("\${kred} = \"0\")?zaves)

exten => 3333,n,Dial(SIP/SPAIN/\${EXTEN})

exten => 3333,n,Congestion()

exten => 33333,n(zaves),Hangup()

[IPPHONE]

exten => 5555,1,NoOp()

exten => 5555,n,Set(kred=\${ODBC_SQL(select cash from kredit where username = \"\${EXTEN}\")})

exten => 5555,n,GotoIf("\${kred} = \"0\")?zaves)

exten => 5555,n,Dial(SIP/ IMSI230026700507478@127.0.0.1:5062)

exten => 5555,n,Congestion()

exten => 5555,n(zaves),Hangup()

Príloha E: Asterisk konfiguračný súbor iax.conf

```
[general]
autokill=yes
port=4569
bandwidth=low
disallow=all
jitterbuffer=yes
calltokenoptional=0.0.0.0/0.0.0.0
maxregexpire=130
encryption=yes
disallow=all
allow=alaw
allow=gsm
allow=g711

register=>SLOVAKIA:trunk@158.196.244.145:4569

[SLOVAKIA]
type=friend
host=dynamic
trunk=yes
secret=trunk
username=SLOVAKIA
qualify=yes
context=internal
deny=0.0.0.0/0.0.0.0
permit=158.196.244.145/255.255.255.255
```