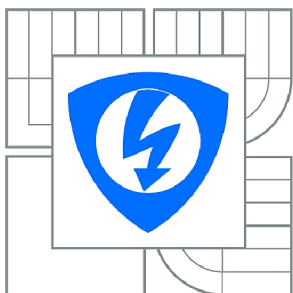




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

EXPERIMENTÁLNÍ MOBILNÍ SÍŤ REALIZOVANÁ POMOCÍ OPEN SOURCE

EXPERIMENTAL MOBILE NETWORK BASED ON OPEN-SOURCE PROJECTS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JAN EFLER

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PAVEL ŠILHAVÝ, Ph.D.

BRNO 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Jan Efler

ID: 138835

Ročník: 3

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Experimentální mobilní síť realizovaná pomocí open source

POKYNY PRO VYPRACOVÁNÍ:

S pomocí open source projektů yateBTS, vývojového rádiového KITu USRP1 a open source PBX realizujte testovací mobilní síť. Nastudujte a ověřte možnosti open source projektů yateBTS, PBX yate, Open VoLTE a Open HSS.

DOPORUČENÁ LITERATURA:

- [1] Clark, C. Software Defined Radio: with GNU Radio and USRP. McGraw-Hill Professional, New York. 2008, ISBN: 978-0071498838.
- [2] Harte, L. Introduction to GSM: Radio Channels, Logical Channels, and Network Operation. Althos Publishing. 2005. ISBN: 978-1932813043.
- [3] YATEBTS – Documentation. [online], [cit. 2014-09-10]. Dostupné z <http://wiki.yatebts.com>.

Termín zadání: 9.2.2015

Termín odevzdání: 26.5.2015

Vedoucí práce: Ing. Pavel Šilhavý, Ph.D.

Konzultanti bakalářské práce:

doc. Ing. Jiří Mišurec, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Tato práce se zabývá realizací experimentální mobilní sítě za využití open source projektů. K realizaci GSM sítě byl využit kit USRP1, na kterém je pomocí GNURadia zprovozněn projekt YateBTS. Pro spojování hovorů je využita ústředna Yate. V práci je popsána instalace a konfigurace sítě

KLÍČOVÁ SLOVA

GSM, YateBTS, Yate, open source, USRP1, GNURadio, Apache

ABSTRACT

This bachelor thesis deals with the realization of an experimental mobile network using open source projects. To implement GSM network was used kit USRP1 which uses YateBTS with GNURadio. To call connection is used PBX Yate. Installation and configuration are part of this thesis.

KEYWORDS

GSM, YateBTS, Yate, open source, USRP1, GNURadio, Apache

EFLER, Jan *Experimentální mobilní síť realizovaná pomocí open source*: bakalářská práce. Místo: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2015. 36 s. Vedoucí práce byl Ing. Pavel Šilhavý, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Experimentální mobilní síť realizovaná pomocí open source“ jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

Místo

.....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Pavlu Šilhavému, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Místo

.....

podpis autora(-ky)



Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
<http://www.six.feec.vutbr.cz>

PODĚKOVÁNÍ

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

Místo

.....

podpis autora(-ky)



EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI



OBSAH

Úvod	8
1 Koncepce softwarového rádia	9
1.1 GNU Radio	9
1.2 Yate	9
1.3 YateBTS	10
1.4 Open VolTE	11
1.5 Open HSS	11
1.6 Kit USRP1	11
2 Instalace testovací sítě	13
2.1 GNU Radio	13
2.2 Pobočková ústředna Yate	16
2.3 YateBTS	17
2.4 Webové rozhraní	17
2.5 Poznátky při realizaci	19
3 Konfigurace GSM sítě	20
3.1 Vyhledání a ověření volného kanálu	20
3.2 Zjištění IMSI čísla	21
3.3 Minimální konfigurace	21
3.4 Webové rozhraní	22
3.5 Odchozí volání	24
4 Spuštění sítě	27
4.1 Virtualizace projektu	28
4.2 Příkazová řádka	29
5 Závěr	32
Literatura	33
Seznam příloh	35
A Obsah přiloženého DVD	36

ÚVOD

Opensource projekty se především využívají k testování zabezpečení daného systému. Cílem je zamezit pozdnímu odhalení bezpečnostních problémů. Další výhodou těchto projektů je cena a rychlost v rekonfiguraci projektu, jelikož z větší části je projekt řešen softwarově.

Tato bakalářská práce se zaměřuje na využití „opensource“ projektů k realizaci mobilní sítě, kdy podrobně popisuje instalaci a nastavení všech potřebných softwarů pro správnou funkčnost. K vytvoření GSM sítě byl použit kit USRP1. Jako ovladač kitu slouží software UHD, který využívá GNURadio k softwarovému vytváření specifických částí rádiového zařízení. Spojování hovorů je realizováno pomocí po-
bočkové ústředny Yate. Hlavním projektem prezentující základnovou stanici s řídicí jednotkou je software YateBTS.

V první kapitole jsou popsány softwary a hardware pro realizaci GSM a LTE sítě, kdy na to navazuje kapitola třetí podrobným popisem nastavení všech potřebných programů. Druhá kapitola se zabývá instalací jednotlivých softwarů potřebných k vytvoření GSM sítě. Poslední kapitola je věnovaná spuštění GSM sítě, jak na fyzickém počítači, tak na virtuálním stroji.

1 KONCEPCE SOFTWAREOVÉHO RÁDIA

Softwarově definované rádio obsahuje vysílač a přijímač, jejichž pracovní parametry lze měnit pomocí změn příslušného softwaru, aniž by došlo ke změnám v hardwarových komponentách, ovlivňující rádiové vysílání. V praxi to znamená, že hlavní funkce rádiového přijímače a vysílače jsou tvořeny hardwarem, který je řízen softwarově. Softwarem lze aktivovat (de)modulátory, filtry, nastavovat frekvenci, aplikovat různé komunikační protokoly, atd. s prakticky okamžitou změnou v systému.

Významný podíl použitelnosti softwarového rádia zaujímají mobilní buňkové systémy GSM/EDGE/UMTS ap. Mezi výrobce zabývající se výrobou kitů pro softwarová rádia patří např. Ettus Research a její kit USRP1 (Universal Software Radio Peripheral), který byl využit v této práci [8] [9].

Nástroje potřebné k vytvoření GSM sítě jsou popsány v podkapitolách 1.1 až 1.3 a poté v podkapitolách 1.4 a 1.5 jsou popsány nástroje LTE sítě.

1.1 GNU Radio

GNU Radio je volný, otevřený zdrojový software poskytující signálové procesní bloky pro softwarová rádia. Je licencován pod GPLv3 licenci (General Public License). Lze ho využít pro operační systémy Windows, Mac a Linux. Aplikace je z větší části napsána v jazyce Python, avšak pro výpočetní část je napsán v jazyce C++ z důvodu využití programu v reálném čase.

K používání GNU Radia není nezbytně nutná znalost v programování, software lze využívat v grafickém prostředí GNU Radio Companion, které se velmi podobá prostředí Simunlinku. Společně s GNU Radiem jsou předpřipraveny různé systémové nástroje a programy pro práci se signály. Jedním z nich je například nástroj *uhd_fft*, který lze využít jako jednoduchý spektrální analyzátor [3].

1.2 Yate

Yate (Yet Another Telephone Engine) je pobočková ústředna nové generace zaměřená na VoIP (Voice over Internet Protocol) a PTSN (Public Switched Telephone Network). Aplikace je napsán v jazyce C++. Podporuje skriptování v různých programovacích jazycích, které byly uvolněny z důvodu usnadnění vývoje externích funkcí ústředny Yate (PHP, Python a Perl knihovny). Je distribuována pod volně šiřitelnou licenci GPLv2 (General Public License) [16].

Ústřednu lze použít jako:

- VoIP server, klient, brána

- Jabber server, klient
- IP Telephony server a/nebo klient
- SIP registrační server, router
- a mnoho další

1.3 YateBTS

YateBTS (Yate Base Transceiver Station) je Unixovou aplikací, vyvinutou rumunskou společností NullTeam, využívající softwarového rádia pro vytvoření přístupového bodu do GSM bezdrátové sítě, která je kompatibilní s 2G a 4G sítěmi. Pro spojování jednotlivých hovorů je zapotřebí pobočková ústředna PBX (Private Branch Exchange). V konceptu YateBTS se využívá vlastní ústředny Yate. Aplikace je napsaná v jazyce C++ a platí na ni otevřené licence AGPLv3 (Affero General Public License) verze 3 a GPLv2 (General Public License) verze 2 [17].

YateBTS od verze 3 disponuje webovým rozhraním. K tomu je potřeba webový server. V našem případě je využit webový server s otevřeným kódem Apache HTTP Server. Instalace a konfigurace je popsána v kapitolách 2.4 a 3.4. Přístup k webovému rozhraní YateBTS vede přes protokol HTTP zadáním příkazu v internetovém prohlížeči <http://127.0.0.1/nib>. Zde se nacházejí tyto sekce:

- Subscribers contains
 - List Subscribers
 - * každé IMSI je možné vložit individuálně
 - * vložení regulárního výrazu pro porovnání IMSI čísel
 - * vše se zapíše do subscribers.conf
 - Country code and SMSC
 - * volací kód země
 - Online Subscribers
 - * seznam všech účastníků, kteří jsou připojeni k síti
 - * aktivní, pokud je spuštěné YateBTS
 - Rejected IMSIs
 - * seznam zakázaných IMSI čísel
 - * aktivní, pokud je spuštěné YateBTS
 - Manage SIMs
 - * seznam všech zapsaných SIM pomocí pySIM
- BTS Configuration
 - formulář pro každou sekci konfiguračního souboru ybts.conf
- Outgoing
 - konfigurace IAX a SIP protokolu pro odchozí volání

1.4 Open VoLTE

VoLTE (Voice over Long Term Evolution) je technologie umožňující přenášet hlasové hovory přes síť LTE. Z technického hlediska jde o IP telefonování. Tato služba má 3krát větší datovou a hlasovou kapacitu než technologie 3G a 6krát větší než technologie 2G. Tím bylo dosaženo větší kapacity pro datové služby. VoLTE je základem pro moderní uživatelské prostředí zahrnující HD Voice (zkvalitnění hlasového hovoru), HD konference, IP zprávy. Technologie byla poprvé uvedena na trh v Jižní Koreji v srpnu 2012. Během dvou let se tato služba rozšířila do Asie a Severní Ameriky, později i do celého světa. VoLTE je založen na stávajícím standardu 3GPP (The 3rd Generation Partnership Project) a vychází z MMTel (multimedia telephony service). Přestože MMTel tvoří základ VoLTE, jeho nedílnou součástí jsou také EPC (Evolved Packet Core) a LTE [14].

1.5 Open HSS

HSS (Home Subscriber Server) je domovský uživatelský server, který obsahuje databázi uživatelů, sloužící entitám IMS (IP Multimedia Subsystem) sítě zajišťující volání. Obsahuje profily uživatelů, provádí ověřování a autorizaci účastníků, poskytuje informace o fyzickém umístění uživatele. Entity komunikující se serverem jsou aplikační servery, které hostí a spouštějí službu v prostředí IMS. Komunikace probíhá pomocí DIAMETER protokolu, jenž vychází z protokolu RADIUS. Tento protokol je součástí řady AAA (autentizační, autorizační a účtovací). Hlavními rozdíly DIAMETER protokolu oproti RADIUS protokolu jsou:

- využívá spolehlivý transportní protokol (TCP)
- může využít zabezpečení na transportní vrstvě
- má větší adresní prostor
- jde o klient-server protokol
- lze použít stavový i bezstavový model
- oznamuje chyby

K zabezpečení využívá protokol IPsec (Internet Protocol Security) nebo TLS (Transport Layer Security) [10].

1.6 Kit USRP1

Universal Software Radio Peripheral je hardware od společnosti Ettus Research, zobrazen na 1.1. Obsahuje 2 duální analogově-digitální konvertory s rychlostí 64 MS/s (milionů snímků za sekundu), 2 duální digitálně-analogové konvertory s rychlostí 128

MS/s. Komunikace počítače a kitu probíhá přes USB port. Pro správnou funkčnost je třeba tento kit připojit k rozhraní USB verze 2.0. USRP1 umožňuje připojit dvě plnohodnotné desky. Toho lze využít pro aplikace, které vyžadují vysokou izolaci vysílací a přijímací části. USRP podporuje systémy Linux, MAC OSX a Windows. [13]

Naše verze je ještě vybavena 52 MHz generátorem hodinového impulzu namísto 64 MHz generátoru hodinového impulzu pro bezproblémové připojení mobilních stanic. Dále je vybavena dvěma deskami RFX900 a anténami VERT900. Kit je využit pro vytvoření mobilní sítě pomocí open source projektů. Komunikaci zprostředkovává software GNU Radio, avšak pouze do verze 3.4.2, kdy skončila podpora ovladače *libusrp*. Tento nedostatek byl vyřešen při instalaci v kapitole 2.1.

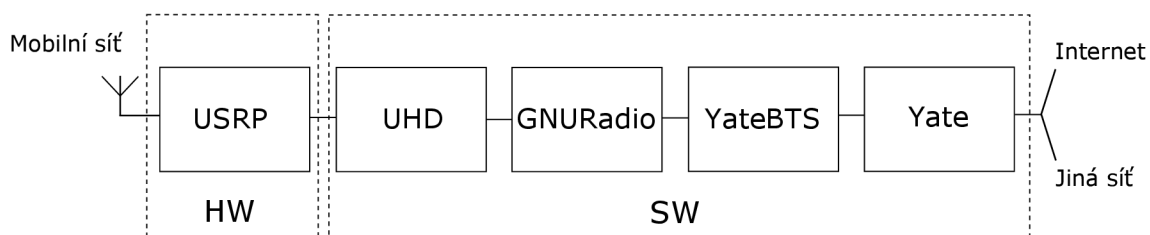


Obr. 1.1: Kit USRP1

2 INSTALACE TESTOVACÍ SÍTĚ

Instalace proběhla v budově T12, učebně C5.34, kde byl pro práci vyhrazen počítač s nainstalovaným operačním systémem Linux v distribuci Ubuntu 12.04 s jádrem 3.7. a 64-bitovou architekturou. Dále byl využit vývojový kit USRP1, který obsahoval dvě desky RFX900, dvě antény VERT900 a generátor hod. impulsu s vysokou stabilitou, neboť původní 64Mhz nebyl pro svou přesnost a hodnotu kmitočtu pro tuto aplikaci vhodný; přihlásit se dokázaly jen některé starší modely. K testování sítě byl použit mobilní telefon LENOVO S750S a starší mobilní zařízení Nokia 3100.

K vytvoření GSM sítě byl užit volně šiřitelný projekt YateBTS a pro práci v grafickém prostředí webový server Apache. Jednotlivé části - jejich instalace a nastavení je popsáno v samostatných kapitolách. Rozmístění a propojení jednotlivých prvků znázorňuje obrázek 2.1. Kit USRP je připojen k počítači pomocí USB rozhraní. Komunikaci zprostředkovává ovladač UHD, který využívá GNURadio k softwarovému vytváření specifických částí rádiového zařízení. Ústřední stanicí s řídicí jednotkou je software YateBTS. Spojování hovorů je realizováno pomocí pobočkové ústředny Yate.



Obr. 2.1: Propojení projektů testovací sítě

Minimální hardwarová konfigurace počítače není nijak specifikována. Jediný požadavek je na USB port verze 2.0. Přidáním 52MHz generátoru hodinového impulsu do kitu USRP nepotřebujeme specifickou konfiguraci pro mobilní stanice. Pouze podpora GSM sítě je vyžadována u mobilních stanic.

2.1 GNU Radio

Instalaci GNU Radia lze realizovat několika způsoby. Problémem je však to, že GNU Radio ke své funkci využívá hned několik knihoven a i se sebelepším návodem, nelze vždy předejít k nečekaným chybám při instalaci. Jednou z cest, jak nainstalovat GNU Radio, je využití pre-kompilovaného balíčku *gnuradio* z repozitářů konkrétní distribuce, avšak u této možnosti je problém v neaktuálnosti verze GNURadia.

Dalším řešením (a pro nás asi nejschůdnějším) je využití kompilačního skriptu od Marcuse Leecha, jenž je dostupný na stránkách GNURadia [3]. Tento skript

nainstaluje nejnovější verzi a provede veškerá nastavení. Skript pomocí Git (systém správy verzí) stáhne software i ovladače potřebné k instalaci. V lednu 2015 byla verze GNURadia 3.7.5.

Před instalací je dobré si vytvořit vlastní složku, kam budeme ukládat celý projekt. Ve spuštěném terminálu se přesuneme do naší složky a skript za pomoci příkazu *wget* uložíme do počítače. Nyní je potřeba nastavit složce práva pro čtení a zápis. Skript si automaticky stáhne potřebné knihovny a ovladače. Pokud máme vše připraveno, můžeme skript spustit [4].

```
$ sudo wget http://www.sbrac.org/files/build-gnuradio
$ sudo chmod a+x ./build-gnuradio
$ sudo ./build-gnuradio
```

Po správné instalaci, která trvá necelou hodinu, nám skript vypíše hlášku **All Done**. Nyní disponujeme nejnovější verzí GNU Radia, v našem případě to je verze 3.7.5. Jak jsem již zmiňoval v předchozí kapitole 1.6, knihovna *libusrp* je podporována pouze do verze 3.4.2, proto je třeba tuto verzi GNURadia snížit. Starší verzi lze získat opět pomocí příkazu *wget* ze stránek GNURadia. Uloženou verzi 3.4.2 je třeba rozbalit.

```
$ sudo wget http://gnuradio.org/redmine/attachments/download/798/gnuradio-3.4.2.tar.gz
$ sudo tar -xf gnuradio-3.4.2.tar.gz
```

Před spuštěním instalace je ještě třeba upravit dva soubory, ve kterých změníme hodnotu použitého generátoru hodinového impulzu. Jedná se o soubor *usrp_basic.cc* a *db_bitshark_rx.cc*. Pro úpravu souboru využijeme například editor *gedit*.

```
$ sudo gedit gnuradio-3.4.2/usrp/host/lib/usrp_basic.cc
```

Zde je třeba upravit řádek 110, na kterém se nachází:

```
d_verbose (false), d_fpga_master_clock_freq(64000000), d_db(2)
```

a přepsat jej na:

```
d_verbose (false), d_fpga_master_clock_freq(52000000), d_db(2)
```

Obdobně to platí i pro druhý soubor.

```
$ sudo gedit gnuradio-3.4.2/usrp/host/lib/db_bitshark_rx.cc
```

Řádek 151 upravíme takto:

```
set_clock_scheme(0,64000000);
set_clock_scheme(0,52000000);
```

Nyní můžeme spustit instalaci GNURadia verze 3.4.2. Nacházíme-li se v adresáři, kam jsme rozbalili GNU Radio, určíme, které součásti mají být změněny při konfiguraci. Poté již následuje kompilace a instalace dané verze. [5].

```
$ cd gnuradio-3.4.2
```

```

$ ./configure --disable-all-components --with-fusb-tech=
  libusb1 --enable-usrp --enable-gruel --enable-gnuradio-
  core --enable-gr-usrp --enable-config --with-boost-libdir
  =/usr/lib
$ make -j<N>
$ make check
$ sudo make install

```

Je možné, že bude vyžadovaná nějaká knihovna či některý balíček potřebný k instalaci. Na chybějící knihovny skript upozorní. V našem případě chyběla knihovna *sdcc*. V novém okně terminálu zadáme níže uvedený příkaz a knihovnu doinstalujeme.

```
$ sudo apt-get install sdcc
```

Po přidání chybějící knihovny či balíčku je nutné opět provést kompilaci. Pokud vše proběhlo v korektně, je nutné ještě upravit cestu k Pythonu v souboru *bashrc*:

```
$ sudo gedit ./bashrc
```

Na konec tohoto souboru je třeba přidat následující řádky:

```
PYTHONPATH=/usr/local/lib/python2.7/dist-packages
```

```
PKG_CONFIG_PATH=/usr/local/lib/pkgconfig
```

Pro správnou funkci je vhodné tento krok ještě potvrdit v systému, zadáním následujících příkazů:

```

$ sudo ldconfig
$ sudo export PYTHONPATH=/usr/local/lib/python2.7/dist-
  packages
$ sudo export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig/

```

Nainstalovanou verzi GNU Radia ověříme pomocí příkazu:

```
$ gnuradio-companion --version
```

Výsledkem příkazu by měla být informace o nainstalované verzi GNU Radia 3.4.2.

```
GNU Radio Companion 3.4.2
```

```

This program is part of GNU Radio
GRC comes with ABSOLUTELY NO WARRANTY.
This is free software,
and you are welcome to redistribute it.

```

Zda je korektně nainstalován i ovladač UHD, potřebný pro komunikaci s kitem, zjistíme zadáním příkazu:

```
$ uhd_find_device
```

Pokud je vše nainstalováno správně, výsledkem budou informace o připojeném kitu[7].

```
linux; GNU C++ version 4.6.3; Boost_104800; UHD_003.008.002-80-ge28d7844
```

```
-- Loading firmware image: /usr/local/share/uhd/images/usrp1_fw.ihx... done
```

```
-----
```

```
-- UHD Device 0
```

```
-----
```

```
Device Address:
```

```
  type: usrp1
```

```
  name:
```

```
  serial: E4R22Y3U
```

2.2 Pobočková ústředna Yate

Před instalací samotného projektu YateBTS je třeba nainstalovat ústřednu Yate, kterou lze získat pomocí SVN (systém pro správu zdrojových kódů). Přepneme se do námi zvoleného adresáře a zadáme příkaz:

```
$ sudo svn checkout http://voip.null.ro/svn/yate/trunk yate
```

Nyní se přepneme do adresáře *yate* a spustíme *autogen.sh*. Tímto vygenerujeme konfigurační skript, podle něhož se nainstaluje ústředna Yate.

```
$ cd yate
```

```
$ ./autogen.sh
```

```
$ ./configure
```

```
$ sudo make install-noapi
```

Místo příkazu *sudo make install-noapi* lze využít pouze *sudo make install*, ale je zapotřebí mít nainstalovaný *doxygen* či *kdoc* balíček.

Následně je třeba provést kompilaci Yate. Pokud se nacházíme v adresáři *yate*, použijeme následující příkazy:

```
$ ./autogen.sh
```

```
$ ./configure
```

```
$ sudo make
```

```
$ sudo make install
```

Tímto máme nainstalovanou pobočkovou ústřednu. Verzi ústředny lze zjistit pomocí příkazu zadaného v okně terminálu [17].

```
$ yate --version
```

2.3 YateBTS

Projekt YateBTS je dostupný ze stránek [17]. Pomocí SVN uložíme zdrojový kód do zvoleného adresáře.

```
$ sudo svn checkout http://voip.null.ro/svn/yatebts/trunk
yatebts
```

V počítači se nesmí nacházet dvě verze aplikace. Podle níže uvedeného příkazu zjistíme, zda se v počítači nenachází starší verze aplikace.

```
$ which -a yate-config
```

```
/usr/local/bin/yate-config
```

Pokud zjistíme přítomnost dvou verzí, je třeba starší verzi odinstalovat. Pro zjištění verze slouží přepínač - *-version*.

```
$ /usr/local/bin/yate-config --version
```

5.3.0

Odinstalaci starší verze provedeme v adresáři, kde je nainstalovaná za použití následujícího příkazu.

```
$ sudo make uninstall
```

Jestliže je v počítači pouze jedna verze, můžeme pokračovat. Přepneme se proto do adresáře, kam jsme uložili YateBTS a spustíme *autogen.sh*, který nám vygeneruje konfigurační skript. Při konfiguraci si skript zkontroluje, zda má nainstalovány všechny potřebné knihovny a balíčky. Pokud by nějaká knihovna chyběla, konfigurace vše zahlásí a je nutné před samotnou instalací tuto knihovnu doinstalovat.

```
$ cd yatebts
$ ./autogen.sh
$ ./configure
$ sudo make install
```

Nyní máme ústřednu Yate a softwarové rádio YateBTS [17].

2.4 Webové rozhraní

Chcete-li nastavovat softwarové rádio pomocí webového rozhraní, je třeba nainstalovat Apache Server. Současně je třeba nainstalovat i podporu PHP a MySQL databázi a vše korektně propojit. [1]

Instalace samotného serveru zabere pouze dva řádky.

```
$ sudo apt-get update
$ sudo apt-get install apache2
```

Správnost instalace lze zkontrolovat pomocí internetového prohlížeče. Do adresného řádku vložíme IP adresu (localhost). Pokud je vše v pořádku, objeví se nám text *It works!*

Nyní je třeba nainstalovat MySQL databázi.

```
$ sudo apt-get install mysql-server libapache2-mod-auth-  
mysql  
$ sudo apt-get install php5-mysql
```

Během instalace budete vyzváni k nastavení hesla.

Je také potřeba aktivovat MySQL databázi:

```
$ sudo mysql_install_db
```

K dokončení instalace je třeba spustit skript, který nastaví databázi. Skript se bude dotazovat na různá nastavení. Nejjednodušší způsob je všechny tyto výzvy potvrdit písmenem *y*, tím bude databáze nastavena do výchozího stavu.

```
$ sudo /usr/bin/mysql_secure_installation
```

Posledním krokem je nainstalovat podporu PHP. Jedná se o webový skriptovací jazyk, který je široce používaný k vytváření dynamických webových stránek. V průběhu je dobré potvrdit všechny výzvy písmenem *y*.

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

Je dobré zanést *index.php* do konfiguračního souboru *dir.conf*. Tento soubor se nachází v */etc/apache2/mods-available/*. Pokud se nacházíme ve složce, využijeme textový editor (např. *gedit*) a soubor upravíme.

```
<IfModule mod_dir.c>  
    DirectoryIndex index.php index.html index.cgi index.pl index.php index.  
    xhtml index.htm  
</IfModule>
```

Pokud máme webový server a ostatní součásti nainstalovány. Pomocí SVN uložíme (do námi zvolené složky) a nakopírujeme složku *nib_web* do adresáře */var/www/html*.

```
$ svn co http://voip.null.ro/svn/yatebts/trunk/nib/web  
nib_web  
$ cd /var/www/html  
$ ln -s /usr/src/nib_web nib
```

Pro správnou funkčnost je třeba ještě přidělit práva pro čtení a zápis složce */usr/local/etc/yate/*

```
$ chmod a+rw /usr/local/etc/yate
```

2.5 Poznátky při realizaci

Při spuštění skriptu od Marcuse Leecha běžným uživatelem došlo k chybové hlášce.

```
sudo: >>> /etc/sudoers: syntax error near line 28 <<<
sudo: parse error in /etc/sudoers near line 28
sudo: no valid sudoers source found, quitting
sudo: unable to initialize policy plugin
```

Tato chybová hláška byla způsobena zablokovaným účtem root. Ani jako uživatele root nás skript daleko nepustil. Skript skončil při instalaci rozhraní UHD. Chyba byla vyřešena tak, že byl odblokován účet root za pomoci příkazu *sudo passwd root*. [11]

Pokud skript skončí chybovou hláškou, je dobré tento skript spustit ještě s parametrem *-v(./build-gnuradio -v)*. Tím získáme podrobnější výpis a můžeme zjistit v jaké fázi byl skript ukončen.

U snižování verze GNU Radia došlo k problému s libboost knihovnou, kterou nemohla konfigurace najít.

```
checking if more special flag are required for pthreads... no
checking whether the boost::thread includes are available... yes
configure: error: Could not link aainst libboost_thread!
```

Toto bylo vyřešeno přidáním příkazu *--with-boost-libdir=/usr/lib* ke konfiguraci instalace [2]

Při pokusu o spuštění softwarového rádia došlo k chybě s knihovnou *libusrp-3.4.2.so.0.0.0*. Chyba byla způsobena špatným umístěním této knihovny. Správné umístění knihoven je */usr/local/lib64*, pokud používáme 64-bitovou architekturu, jestliže používáme systém s 32-bitovou архитектурou, správné umístění knihoven je */usr/local/lib*.

Při spuštění projektu nastala chyba s vysílačem.

```
Transceiver exiting with status 134
```

V adresáři */usr/local/share/usrp/rev4* z neznámých důvodů chyběla knihovna *std_inband.rbf*. Tuto knihovnu lze nakopírovat z adresáře */usr/local/lib/yate/server/bts*.

3 KONFIGURACE GSM SÍTĚ

K samotnému běhu je třeba ještě nastavit konfigurační soubory ústředny a softwarového rádia. Pomocí softwaru kalibrate-uhd a kitu USRP1 vyhledáme volný kanál ve vysílaném pásmu, aby nedocházelo k rušení rádia. Pro registraci uživatelů je třeba znát IMSI číslo použité SIM karty.

3.1 Vyhledání a ověření volného kanálu

Aby rádio fungovalo tak, jak má, a bylo co nejméně rušeno, je zapotřebí nalézt volný kanál ve vysílacím pásmu. Obsazenost kanálů v místě vysílání lze zjistit pomocí stránek gsmweb.cz, které obsahují informace o všech obsazených kanálech. Bylo zjištěno, že kanál 51 je volný. Pro potvrzení volného kanálu byl využit program kalibrate-uhd. Tento program během svého běhu, za pomoci USRP rádia, testuje frekvence a kanály ve zvoleném GSM pásmu. Pokud na dané frekvenci již nějaká stanice vysílá, aplikace vypíše o jaký kanál se jedná a na jaké frekvenci se nachází. Takto projede celé zvolené pásmo. Z výpisu je patrné, na kterém kanále se vysílá. Pokud výpis námi zvolený kanál neobsahuje, lze ho využít pro naši testovací síť. Program kalibrate-uhd je dostupný ze stránek [6]. Program uložíme do adresáře, rozbalíme a zkompilujeme.

```
$ sudo wget http://ttsou.github.com/kalibrate-uhd/kal-v0
.4.1.tar.bz2
$ sudo tar -xf kal-v0.4.1.tar.bz2
$ cd kal-v0.4.1
$ ./bootstrap
$ ./configure
$ sudo make
```

Aplikaci kalibrate-uhd spustíme pomocí příkazu uvedeného níže. Vstupním parametrem je GSM pásmo.

```
$ ./kal -s GSM900
```

```
kal: Scanning for GSM-900 base stations.
```

```
GSM-900:
```

```
chan: 21 (939.2MHz - 17.715kHz) power: 3382.20
chan: 22 (939.4MHz - 17.699kHz) power: 1194.89
chan: 37 (942.2MHz + 32.268kHz) power: 1741.49
chan: 47 (944.4MHz + 39Hz) power: 2583.52
chan: 59 (946.8MHz + 33Hz) power: 4678.74
chan: 84 (951.8MHz + 49Hz) power: 12251.27
```



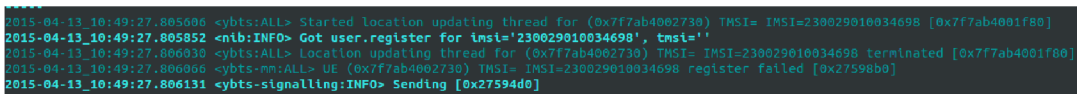
```
chan: 86 (952.2MHz - 17.696kHz) power: 7198.34
chan: 87 (952.4MHz + 17.692kHz) power: 3240.29
chan: 102 (955.4MHz + 32.407kHz) power: 1579.80
chan: 103 (955.6MHz + 32.261kHz) power: 1533.35
```

3.2 Zjištění IMSI čísla

Pro registraci uživatelů do sítě je třeba znát IMSI číslo SIM karty. Toto číslo se dá zjistit za pomoci softwaru pro analýzu mobilního zařízení například MobileEdit, avšak tento program je placený. Nebo můžeme využít naše nainstalované rádio a kit USRP1. Pokud máme v konfiguračním souboru `ybts.conf` 3.3 nastaveno v jakém pásmu, na jakém kanále a jaký vysílač používáme, lze IMSI číslo zachytit při připojení do naší testovací sítě. Po spuštění testovací sítě s maximální možnou úrovní výpisu příkazem:

```
$ sudo yate -vvvvvDo
```

lze zachytit pokus o připojení naší SIM karty do sítě obrázek 3.1.



```
2015-04-13 10:49:27.805606 <ybts:ALL> Started location updating thread for (0x7f7ab4002730) TMSI= IMSI=230029010034698 [0x7f7ab4001f80]
2015-04-13 10:49:27.805852 <nib:INFO> Got user_register for imsi='230029010034698', tmsi=''
2015-04-13 10:49:27.806039 <ybts:ALL> Location updating thread for (0x7f7ab4002730) TMSI= IMSI=230029010034698 terminated [0x7f7ab4001f80]
2015-04-13 10:49:27.806066 <ybts-mm:ALL> UE (0x7f7ab4002730) TMSI= IMSI=230029010034698 register failed [0x27598b9]
2015-04-13 10:49:27.806131 <ybts-signalling:INFO> Sending [0x27594d0]
```

Obr. 3.1: Zjištění IMSI čísla naší SIM

I když registrace neproběhne v pořádku, jelikož ještě nemáme zadané parametry v konfiguračním souboru `subscribers.conf`, lze takto zjistit IMSI číslo naší SIM karty a toto číslo si poznamenat a později část tohoto čísla využít jako regulární výraz pro registraci uživatelů do testovací sítě.

3.3 Minimální konfigurace

Abychom mohli zrealizovat naši síť, je potřeba nastavit několik parametrů v různých konfiguračních souborech ústředny a rádia. Veškeré konfigurační soubory, které budeme upravovat, se nacházejí v `/usr/local/etc/yate/`. V první řadě se musíme rozhodnout, v jakém módu budeme yate spouštět. Na výběr jsou dva; první se jmenuje NIB (Network In Box) a druhý Roaming. NIB mód ke svému chodu využívá již přednastavený javascript, kdežto v roaming módu se musí vše nastavit ručně v konfiguračních souborech. Ke správnému chodu je zapotřebí nastavit, na jakém kanále a v jakém pásmu budeme vysílat. Důležitým parametrem je i jaký vysílač se bude

využívat. Ke správnému běhu Yate je distributorem doporučováno nainstalovat vysílač bladeRF. My však budeme využívat vysílač UHD, který je plně kompatibilní s naším kitem.

Jak jsem psal výše, je třeba zvolit si kanál a pásmo, ve kterém budeme vysílat. Zadání kanálu a pásma se provede v konfiguračním souboru *ybts.conf*. V sekci [gsm] zadáme hodnotu *Radio.Band*(řádek 13), jež nám určuje pásmo, ve kterém budeme vysílat a *Radio.CO*(řádek 22) určuje kanál, v našem případě 51. Dále v tomto dokumentu v sekci [transceiver] odkomentujeme a změníme řádek 159, kde se nachází výběr vysílače. Pro výběr módu, který budeme používat, slouží řádek 759.

```
Radio.Band=900  
Radio.CO=51  
Path=./transceiver-usrp1  
mode=nib
```

Při použití módu NIB je třeba nastavit konfigurační soubor *subscribers.conf*. Zde máme na výběr, jak se budou registrovat uživatelé do sítě. Jednou z možností je využít regulárního výrazu pro kontrolování IMSI čísel a druhou možností je nastavení pro každého uživatele individuálně. Při individuálním nastavení potřebujeme znát Ki číslo (Authentication key) SIM karty. Pokud používáme SIM kartu, přidělenou našim operátorem, nelze toto číslo v současnosti zjistit z důvodu bezpečnosti. Jelikož testování sítě proběhlo na SIM kartě od operátora, nemohli jsme proto číslo Ki zadat. Proto bylo využito regulárního výrazu. Při použití regulárního výrazu nám postačí velice jednoduchý výraz. Pokud se shodují první tři číslice IMSI čísla, dojde k povolení registrace. Ostatní IMSI čísla budou zamítnuta. Je zde potřeba také vyplnit volací kód země. V našem případě 420.

```
country_code=420  
regexp=^230
```

Nyní se budeme věnovat konfiguračnímu souboru *javascript.conf*. Zde v sekci [general] změníme řádek 15.

```
routing=welcome.js
```

3.4 Webové rozhraní

Zadáním příkazu *http://127.0.0.1/nib* do adresního řádku prohlížeče můžeme nastavit softwarové rádio pomocí webového rozhraní. V záložce BTS Configuration v sekci GSM nastavíme pásmo a kanál, ve kterém budeme vysílat. Ostatní hodnoty ponecháme ve výchozím stavu. Potvrdíme tlačítkem Submit obr. 3.2.

Obr. 3.2: Nastavení kanálu a vysílacího pásma

Přepneme se do záložky Transceiver, kde vybereme vysílač. V našem případě usrp-1. Zbylé hodnoty můžeme nechat nezměněné. Hodnoty se zapíše do konfiguračního souboru *ybts.conf* obr. 3.4.

Obr. 3.3: Volací kód České republiky

V záložce Subscribers využijeme regulárního výrazu na porovnávání IMSI čísel. To je důležité při registraci uživatelů do sítě. Po vložení regulárního výrazu, který může vypadat například takto: $\wedge 230$, bude povolena registrace uživatelům, jejichž IMSI čísla SIM karty začínají číslem 230. Uložíme je tlačítkem Save obr. 3.5. Regulární výraz byl zapsán do konfiguračního souboru *subscribers.conf*.

Subscribers | **BTS Configuration** | Outgoing

GSM | GPRS | Control | **Transceiver** | Tapping | Test | YBTS

Set parameters values for section [transceiver] to be written in ybts.conf file.

Path	<input type="text" value="./transceiver-usrp1"/>	?
Args	<input type="text"/>	?
MinimumRxRSSI	<input type="text" value="-63"/>	?
RadioFrequencyOffset	<input type="text" value="Factory calibrated"/>	?
TxAttenOffset	<input type="text" value="Factory calibrated"/>	?
Timeout.Clock	<input type="text" value="10"/>	?

Submit Reset

Obr. 3.4: Nastavení vysílače

Subscribers | **BTS Configuration** | Outgoing

List Subscribers | Country Code and SMSC | Online Subscribers | Rejected IMSIs | Manage SIMs

Regular expression based on which subscriber IMSI are accepted for registration

Regexp*	<input type="text" value="^230"/>	?
---------	-----------------------------------	---

Save Reset

Obr. 3.5: Nastavení regulárního výrazu

V záložce Country Code and SMSC vyplníme volací kód České republiky, tedy 420 a uložíme obr. 3.3.

3.5 Odchozí volání

Dosavadní nastavení nám umožní pouze volání v rámci lokální sítě. Volání z lokální sítě do jiné probíhá přes protokol SIP (Session Initiation Protocol) či IAX (Inter Asterisk Exchange). Oba tyto protokoly se využívají pro přenos signalizace v internetové telefonii. IAX protokol je navržen pro přímé propojení dvou ústředn Asterisk. V našem případě bylo využito protokolu SIP.

Zřízení účtu probíhá v konfiguračním souboru *accfile.conf* naleznete v */usr/local/etc/yate*. Testování proběhlo na školní experimentální síti v laboratoři s připojením k PBX Asterisk v lokální síti.

```

[outbound]          ;nazev uctu
enable=yes           ;akceptovani uctu
username=jmeno       ;jmeno SIP server
password=heslo       ;heslo SIP serveru
server=10.10.10.1    ;IP adresa serveru
protocol=sip         ;vybrany protokol

```

Pro testování byl využit softwarový klient Zoiper a IP telefon. Pro identifikaci softwarového klienta v ústředně Yate, je třeba vytvořit účet v konfiguračním souboru *regfile.conf* umístěn */usr/local/etc/yate*.

```

[uzivatelske jmeno]
password=heslo
napr: [user_400]
password=400

```

V Zoiperu je třeba založit nový účet. To se provádí v menu, nastavení a vytvořit účet. V nově otevřeném okně vybereme protokol (v našem případě SIP) a potvrdíme tlačítkem další 3.6a. Na další stránce je potřeba vyplnit uživatelské jméno (stejně jako v *regfile.conf*), heslo (stejně jako v *regfile.conf*) a doménu (IP adresa počítače). Vše potvrdíme dvakrát stisknutím tlačítka další 3.6b. Nyní se bude účet snažit připojit k ústředně. Pokud máme ústřednu spuštěnou, dojde k registraci softwarového klienta do ústředny. Pokud je vypnutá, odložíme registraci na později zaškrtnutím třetí možnosti „Vím co dělám, ulož tyto informace“ 3.6c. V nastavení účtu (Nastavení -> Předvolby -> Pokročilý) zakážeme službu STUN (Session Traversal Utilities for NAT) 3.6d.

Volací plány jsou uloženy v konfiguračním souboru *regextroute.conf* (v sekci [default]), který se nachází v */usr/local/etc/yate*. Zapisují se pomocí regulárních výrazů.

```

cislo=sip/sip:user@IP:port
napr: ^603=sip/sip:user_400@192.168.20.125

```

Javaskript *niib* ve výchozím stavu nekouká na volací plány, jelikož všechny moduly jsou nastaveny na stejnou prioritu. Snížením hodnoty priority, v konfiguračních souborech, upřednostníme konkrétní modul. Nejprve snížíme hodnotu *preroute* a *route* v sekci [*priorities*] u konfiguračního souboru *regextroute.conf*.

```

preroute=10
route=10

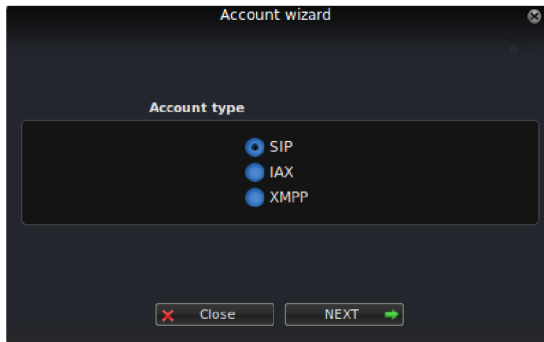
```

Prioritu je třeba změnit i u konfiguračního souboru *regfile.conf*, nacházejícího se v */usr/local/etc/yate*. V sekci [general] na řádce 19 změníme prioritu *call.route*.

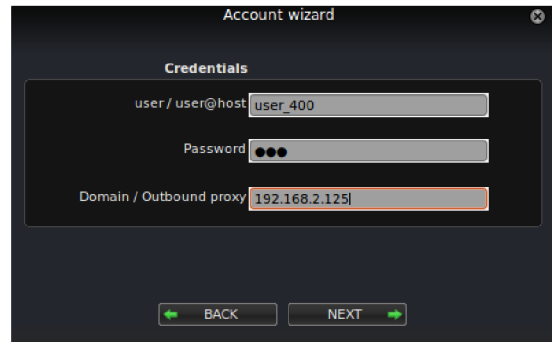
```

route=10

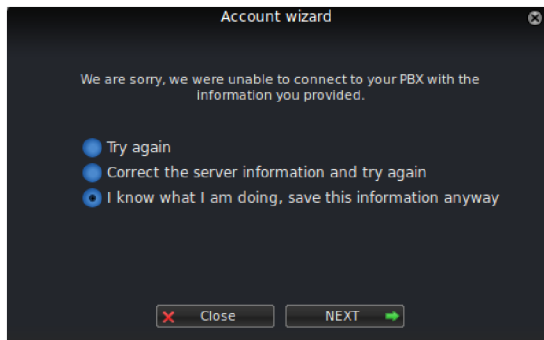
```



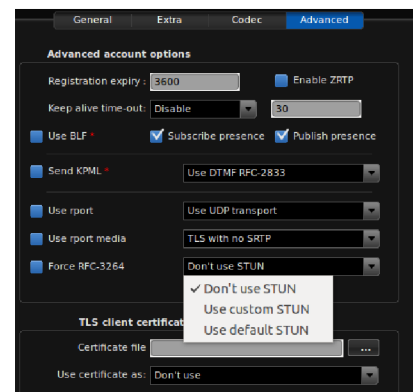
(a)



(b)



(c)



(d)

Obr. 3.6: a) Vybrání protokolu; b) Vytvoření účtu, c) Uložení účtu bez registrace, d) Zakázání služby STUN

Pro odchozí volání je vše nastavené. U projektu byl problém IP telefonu s převedením gsm kodeku. Jelikož IP telefon tímto kodekem nedisponuje, při volání nebylo nic slyšet. Tento problém zahlásila i ústředna.

```
DataTranslator::attachChain 'mulaw' -> 'gsm' failed
```

V počítači chyběla nainstalovaná knihovna libgsm1-dev. Po instalaci knihovny se musela provést reinstalace ústředny a softwarového rádia. Knihovnu nainstalujeme pomocí níže uvedeného příkazu. Reinstalace ústředny a rádia je stejná jako instalace těchto komponentů popsaná v kapitolách 2.2 a 2.3. Při reinstalaci nedošlo ke změně konfiguračních souborů.

```
$ sudo apt-get install libgsm1-dev
```

4 SPUŠTĚNÍ SÍTĚ

Pokud je vše nainstalováno a nastaveno, můžeme spustit testovací síť. Projekt se spouští příkazem:

```
$ sudo yate
```

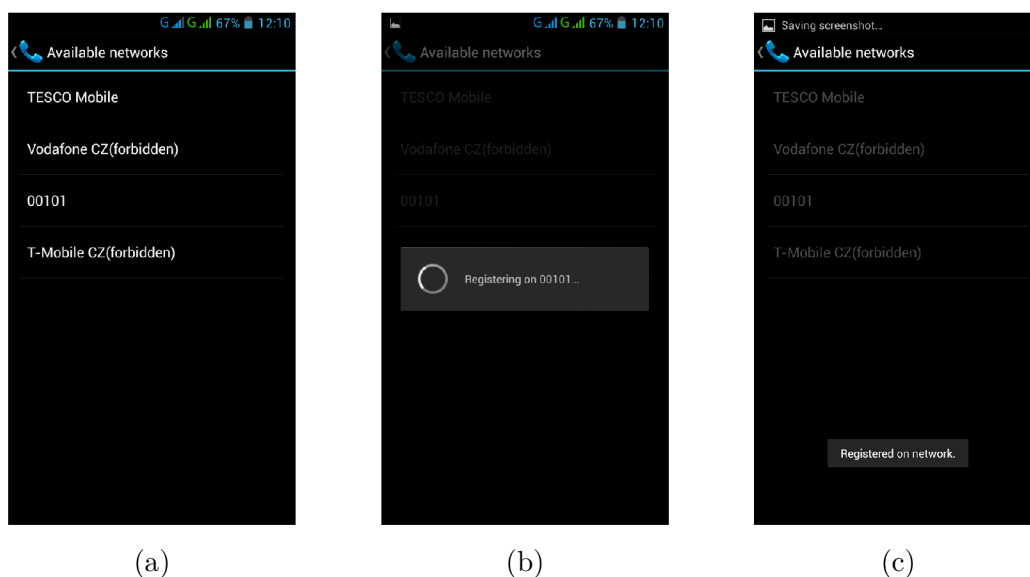
Pokud chceme mít podrobnější přehled o chování aplikace, je třeba spustit aplikaci s parametrem verbose. Tento parametr má několik úrovní. To lze měnit přidáním nebo ubráním parametru v. S maximálním výpisem bude příkaz vypadat následovně:

```
$ sudo yate -vvvvv
```

Je ještě možné příkaz doplnit o parametr -Do, který nám podbarví řádky užitím ANSI kódování

```
$ sudo yate -vvvvvDo
```

Jelikož aplikaci spouštíme s příkazem sudo, budeme vyzváni k vložení hesla. Po spuštění projektu dojde k inicializaci a následně je možné tuto síť dohledat v mobilní stanici. Zde se naše testovací síť hlásí pod číselným názvem 00101. Registrace je zobrazena na snímcích 4.1a, 4.1b a 4.1c.



Obr. 4.1: a) Dostupné sítě; b) Registrování do sítě; c) Zaregistrováno v síti.

Pokud máme projekt spuštěn v maximálním možném výpisu, bude informace o úspěšné registraci vypadat podobně jako na obrázku 4.2.

Vzápětí nám přijde SMS zpráva o úspěšné registraci do naší sítě. Ve zprávě je nám také přidělené číslo, které se skládá z předčíslí pro Českou republiku a posledních

```

-----
2015-04-13_10:44:17.099241 <ybts:ALL> Started location updating thread for (0x7f7ab40023b0) TMSI= IMSI=230030114104122 [0x7f7ab40024f0]
2015-04-13_10:44:17.099511 <nrb:INFO> Got user.register for imsi='230030114104122', tmsi=''
2015-04-13_10:44:17.099694 <nrb:INFO> Allocated random number
2015-04-13_10:44:17.100857 <nrb:INFO> Registered imsi 230030114104122 with number 4204104122
2015-04-13_10:44:17.100918 <ybts:ALL> Location updating thread for (0x7f7ab40023b0) TMSI= IMSI=230030114104122 terminated [0x7f7ab40024f0]
2015-04-13_10:44:17.100957 <ybts-rm:ALL> UE (0x7f7ab40023b0) registered TMSI '' -> '007b0007', IMSI '230030114104122' -> '230030114104122' conn=1 [0x27598b0]
2015-04-13_10:44:17.100987 <ybts-rm:ALL> UE (0x7f7ab40023b0) TMSI=007b0007 IMSI=230030114104122 register succeeded [0x27598b0]
2015-04-13_10:44:17.101082 <ybts-signalling:INFO> Sending [0x27594d0]
-----

```

Obr. 4.2: Úspěšná registrace do mobilní sítě

sedmi čísel našeho IMSI čísla. Telefonní číslo lze změnit v konfiguračním souboru *tmsidata.conf* umístěném v */usr/local/etc/yate*, kam si projekt ukládá všechny doposud registrované uživatele.

Soubor se skládá ze dvou částí; první sekce [tmsi] je poslední použité číslo tmsi (dočasné identifikující číslo). Ve druhé sekci [ues] se nachází hodnoty přidělené IMSI číslu. Řetězec se skládá z čísla tmsi; imei (unikátní číslo přidělené výrobcem telefonu); msisdn (jednoznačné číslo identifikující účastníka); expires (hodnota, kdy dojde k vymazání účastníka z databáze v UNIXovém čase). Nás zajímá číslo msisdn. Pokud chceme změnit přidělené číslo, stačí tuto hodnotu přepsat. Pokud takto číslo přepíšeme (například číslem 200) a uložíme soubor, při dalším přihlášení do sítě již budeme mít telefonní číslo 200.

Aplikace po určitém čase hodnoty přidělené IMSI číslu v konfiguračním souboru smaže. Čas, po který mají být údaje o registrovaných uživateli udržovány, lze nastavit v konfiguračním souboru *ybts.conf*. Výchozí hodnota je nastavena na 10 dní. Maximální doba, po kterou si ústředna bude pamatovat registrované uživatele, je 30 dní. Pro změnu se tedy přepneme do adresáře, kde jsou všechny konfigurační soubory. V našem případě */usr/local/etc/yate* a provedeme změnu v konfiguračním souboru *ybts.conf*.

```

$ cd /usr/local/etc/yate
$ sudo gedit ybts.conf

```

Zde v souboru změním v sekci [ybts] řádek 917. Hodnota se udává v sekundách. Tedy pro dobu 30 dní je třeba zadat číslo 2592000 a například pro dobu 15 dní 1296000.

```
tmsi_expire=hodnota
```

4.1 Virtualizace projektu

Celý tento projekt byl také vyzkoušen na virtuálním stroji. K virtualizaci byl využit software VMware Player 7.0.0. Zde byl nainstalován systém Ubuntu 12.04 v 32-bitové architektuře [12]. Instalace softwaru pro virtualizaci počítačů se provede pomocí instalátoru, který je dostupný z [15]. Následuje vytvoření a nastavení nového virtuálního počítače pomocí ISO obrazu. Po nainstalování virtuálního počítače lze

přejít k instalaci jednotlivých systémů (GNURadio, Yate, YatePBX). Ta proběhla stejně jako na školním počítači, popsaná v kapitole 2. Avšak celá instalace na virtuálním stroji zabrala poměrně více času.

Pro zjištění, zda připojený kit komunikuje s virtuálním strojem, je dobré využít nástroj `uhd_find_devices`. Zde docházelo k delší odezvě, než co kit USRP1 začal komunikovat. Proto je vhodné po každém připojení kitu USRP1 využít tento nástroj. Je možné, že bude vyžadováno ruční připojení USB. Na nástrojové liště VMware Playeru, klikneme pravým tlačítkem myši na USB Device a potvrdíme příkaz Connect (připojit). Nastavení a spuštění projektu je stejné jako v kapitole 3.

Pro realizaci odchozích volání je nutné se připojit k vytvořenému SIP serveru. Nejjednodušším způsobem je využít přímé připojení do lokální sítě a ve virtuálním stroji nastavit síťové připojení na možnost Bridged (přemostění). To se provede v nastavení virtuálního stroje: Síťový adaptér -> Síťové připojení -> Přemostění. Přemostěním připojení se vyhneme dalšímu překladu adres (NAT) a získáme IP adresu lokální sítě.

4.2 Příkazová řádka

Slouží pro ovládání ústředny. Spouští se z příkazové řádky operačního systému, příkazem `telnet IP adresa port`. Ve výchozím stavu ústředna „naslouchá“ na IP adrese 127.0.0.1 a portu 5038. Nastavení lze měnit v konfiguračním souboru `rmanager.conf` umístěn v `/usr/local/etc/yate/`. Konzole pro ovládání rádia byla zavedena od verze 2. Zde lze ovládat ústřednu či filtrovat úrovně výpisu jednotlivých modulů ústředny.

Připojení Pro komunikaci je dobré otevřít si nové okno terminálu a zadat příkaz `telnet 127.0.0.1 5038`. To nám umožní komunikovat s aplikací. Seznam všech příkazů lze vypsát stisknutím klávesnice Tab. Některé příkazy budou rozebrány níže.

- quit
- echo
- auth password
- status
- uptime
- machine
- output
- color
- debug
- meminfo
- drop
- call chan target
- control chan
- reload
- restart
- stop
- alias
- module
- events
- logview
- status jabber
- jabber drop
- jabber create
- jabber debug
- javascript
- external
- filetransfer
- users
- callgen
- sigdump
- accounts
- ybts
- mbts
- cache
- nib

debug Pro nastavení úrovně výpisu jednotlivých modulů se využívá příkazu *debug*. Pomocí něj lze nastavovat jak globálně, tak i jednotlivé části úrovně výpisu. Maximální možná úroveň výpisu je 10, minimální 3. Pokud si chceme vypsat pouze aktuální úroveň, stačí napsat příkaz *debug* a název modulu.

```
debug modul level hodnota  
napr: debug sip level 8
```

ybts Tento příkaz slouží k zastavení, spuštění a restartování softwarové rádia.

```
ybts stop
```

Dočasně vypne softwarové rádio obr. 4.3

```
-----  
2015-04-21_14:00:06.146400 <ybts:ALL> Scheduled stop in 0ms  
2015-04-21_14:00:07.000030 <ybts:ALL> Stopping ...  
2015-04-21_14:00:07.005014 <ybts-signalling:ALL> Worker thread terminated [0xaaa5e0]  
2015-04-21_14:00:07.005143 <ybts-signalling:INFO> Stopped [0xaaa5e0]  
2015-04-21_14:00:07.009927 <ybts-media:ALL> Worker thread terminated [0xaaa300]  
2015-04-21_14:00:07.010238 <ybts-media:INFO> Stopped [0xaaa300]  
2015-04-21_14:00:07.010272 <ybts:NOTE> Peer pid 6195 has not exited - we'll kill it  
2015-04-21_14:00:07.107920 <ybts:INFO> Peer pid 6195 terminated  
2015-04-21_14:00:07.107988 <ybts-command:INFO> Stopped [0xaaa1a0]  
2015-04-21_14:00:07.111248 <transceiver:ALL> Worker thread terminated [0xaa9d20]  
2015-04-21_14:00:07.113100 <transceiver:INFO> Stopped [0xaa9d20]  
2015-04-21_14:00:07.113899 <mbts:ALL> Worker thread terminated [0xaa9f50]  
2015-04-21_14:00:07.118214 <mbts:INFO> Stopped [0xaa9f50]  
2015-04-21_14:00:07.118263 <ybts:NOTE> State changed RadioUp -> Idle  
2015-04-21_14:00:07.118290 <ybts:NOTE> Stopped, waiting for command to start  
2015-04-21_14:00:11.004685 <sip:ALL> Sending UDP keepalive to 192.168.10.5:5060 for 'outbound'
```

Obr. 4.3: Odezva na příkaz *ybts stop*

```
ybts start
```

Zapne softwarové rádio obr. 4.4

```
2015-04-21_14:00:32.122254 <ybts:NOTE> State changed Idle -> Starting restart counter 1/10  
2015-04-21_14:00:32.122362 <transceiver:ALL> Started worker thread [0xaa9d20]  
2015-04-21_14:00:32.122404 <transceiver:INFO> Started [0xaa9d20]  
2015-04-21_14:00:32.122473 <mbts:ALL> Started worker thread [0xaa9f50]  
2015-04-21_14:00:32.122505 <mbts:INFO> Started [0xaa9f50]  
2015-04-21_14:00:32.122560 <ybts-command:INFO> Started [0xaaa1a0]  
2015-04-21_14:00:32.122629 <ybts-signalling:ALL> Started worker thread [0xaaa5e0]  
2015-04-21_14:00:32.122659 <ybts-signalling:INFO> Started [0xaaa5e0]  
2015-04-21_14:00:32.122712 <ybts-media:ALL> Started worker thread [0xaaa300]  
2015-04-21_14:00:32.122743 <ybts-media:INFO> Started [0xaaa300]  
2015-04-21_14:00:32.122776 <ybts:ALL> Starting peer '/usr/local/lib/yate/server/bts/mbts' '(null)'  
2015-04-21_14:00:32.123526 <ybts:INFO> Started peer pid=6308  
2015-04-21_14:00:32.123590 <ybts:NOTE> State changed Starting -> WaitHandshake  
2015-04-21_14:00:32.123632 <ybts:ALL> Restart scheduled in 12000ms [0x7f6f999c7560]
```

Obr. 4.4: Odezva na příkaz *ybts start*

reload Pokud změníme konfigurační soubor, nemusíme restartovat celý projekt. K tomu slouží příkaz *reload*. Tento samotný příkaz načte znovu všechny moduly. Za příkaz *reload* lze přidat název změněného konfiguračního souboru. Dojde k znovunačtení tohoto modulu.

```
reload modul  
napr: reload accfile
```

nib Pokud chceme znát všechny zaregistrované uživatele, využijeme příkazu *nib*. Tento příkaz přečte soubor *tmsidata.conf* a vypíše do konzole registrované IMSI čísla a jejich MSISDN. Pomocí tohoto příkazu je možné zachytit i komunikaci při posílání SMS zpráv.

```
nib list registered
```

IMSI	MSISDN
-----	-----
230029021548632	100
230029010034698	200

```
nib list sms
```

FROM_IMSI	FROM_MSISDN	TO_IMSI	TO_MSISDN
-----	-----	-----	-----
230029021548632	100	230029010034698	200

5 ZÁVĚR

Cílem této bakalářské práce bylo vytvořit experimentální mobilní síť za použití volně dostupných projektů. V teoretické části byly rozebrány opensource projekty, využitě k realizaci mobilní GSM sítě 1.

V praktické části došlo k realizaci mobilní GSM sítě za pomoci kitu USRP1 a open-source softwarů. Hlavním softwarem byl YateBTS, který slouží jako řídicí jednotka. Pro spojení kitu a projektu YateBTS slouží software GNURadio určený pro softwarová rádia. Posledním použitým softwarem při realizaci GSM sítě, byla pobočková ústředna Yate, sloužící k propojení hovorů 2

Instalaci experimentální sítě provázela řada problémů. Ať už to je nedokonalá, či dokonce neexistující dokumentace jednotlivých softwarů, což bývá problém řady open-source projektu, tak i kompatibilita operačního systému Linux. Veškeré neduhy a problémy byly popsány v kapitole 2.5.

Nastavení jednotlivých konfiguračních souborů pro správnou funkci vytvořené GSM sítě byla popsána v kapitole 3. Celá konfigurace GSM sítě byla také navíc odzkoušena na virtuálním stroji, za použití softwaru VMware Player, kapitola 4.

Hlavními klady při využívání open-source projektů jsou nízká cena, bezpečnost, variabilita. Jelikož je volně dostupný zdrojový kód, lze tyto projekty s určitou znalostí libovolně upravovat a modifikovat.

Ve výsledku práce se podařilo realizovat GSM síť pomocí výše zmíněných open-source nástrojů. Tohoto postupu by bylo možné v budoucnu využít například, jako návod pro vytvoření podnikové GSM sítě.

Na tuto práci by bylo možné navázat realizací LTE sítě, například pomocí projektu OpenLTE. Předpokladem pro zprovoznění této sítě je programovatelná SIM karta a mobilní zařízení podporující LTE síť.

LITERATURA

- [1] Apache Server *How To Install* [online]. [cit. 13. 4. 2015]. Dostupné z URL: <<https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu>>.
- [2] *BenV's notes* [online]. poslední aktualizace 27. 3. 2015 [cit. 9. 4. 2015]. Dostupné z URL: <<http://notes.benv.junerules.com/blast-its-boost-slackware64-and-deluge-libtorrent-rasterbar/>>.
- [3] GNURadio *WikiStart - gnuradio.org* [online]. 2006-2013, [cit. 11. 3. 2015]. Celý web. Dostupné z URL: <<http://gnuradio.org>>.
- [4] GNURadio *Installing GNU Radio From Source* [online]. 2006-2013, [cit. 9. 4. 2015]. Dostupné z URL: <<http://gnuradio.org/redmine/projects/gnuradio/wiki/InstallingGRFromSource>>.
- [5] GNU Radio *A Short Guide to Using Libusrp(2) with GNU Radio 3.5.0 and up* [online]. 2006-2013, [cit. 9. 4. 2015]. Dostupné z URL: <http://gnuradio.org/redmine/projects/gnuradio/wiki/UsingLibusrpWith3_5>.
- [6] *Kalibrate UHD* [online]. [cit. 13. 4. 2015]. Dostupné z URL: <<http://ttsou.github.com/kalibrate-uhd/kal-v0.4.1.tar.bz2>>.
- [7] KILIAN, Petr *Analýza GSM sítě pomocí open source SW rádia: diplomová práce*. Brno: Vysoké učení technické, Fakulta elektrotechniky a komunikačních technologií, 2013. 76, 3. Vedoucí diplomové práce Pavel Šílhavý.
- [8] SDR Radio *Co je vlastně SDR* [online]. 2008 - 2015, poslední aktualizace 1. 11. 2013 [cit. 22. 4. 2015]. Dostupné z URL: <<http://sdr.ipip.cz/>>.
- [9] SEHNAL, Jiří *Softwarové rádio* [online]. [cit. 22. 4. 2015]. Dostupné z URL: <<http://coptel.coptkm.cz/index.php?action=2&doc=10889>>.
- [10] OpenHSS *home-subscriber-server.html* [online]. [cit. 9. 4. 2015]. Dostupné z URL: <<http://www.oracle.com/au/products/database/home-subscriber-server-083038.html>>.
- [11] Ubuntu.cz *Wiki Ubuntu* [online]. poslední aktualizace 12. 5. 2014 [cit. 11. 4. 2015]. Dostupné z URL: <http://wiki.ubuntu.cz/root_sudo>.
- [12] Ubuntu 12.04.5 LTS *Precise Pangolin* [online]. [cit. 28. 4. 2015]. Dostupné z URL: <<http://releases.ubuntu.com/12.04/>>.

- [13] *USRP1 Bus Series* [online]. 2015, [cit. 13. 4. 2015]. Dostupné z URL: <http://www.ettus.com/content/files/07495_Ettus_USRP1_DS_Flyer_HR.pdf>.
- [14] *VoLTE The Secret Value of VoLTE* [online]. [cit. 9. 4. 2015]. Dostupné z URL: <<http://blog.tmcnet.com/next-generation-communications/2014/04/the-secret-value-of-volte.html>>.
- [15] *VMware Player download* [online]. [cit. 28. 4. 2015]. Dostupné z URL: <<https://my.vmware.com/web/vmware/downloads>>.
- [16] *Yate Yate/Main* [online]. poslední aktualizace 14. 7. 2014 [cit. 23. 3. 2015]. Dostupné z URL: <<http://old.yate.ro/pmwiki/>>.
- [17] *YateBTS About YateBTS - YateBTS* [online]. poslední aktualizace 23. 2. 2015 [cit. 11. 3. 2015]. Celý web. Dostupné z URL: <http://wiki.yatebts.com/index.php/Main_Page>.

SEZNAM PŘÍLOH

A Obsah přiloženého DVD

36

A OBSAH PŘILOŽENÉHO DVD

Přiložené DVD obsahuje elektronickou verzi této práce (efler_bakalarskaprace.pdf)

Je zde adresář „*konfiguracni_soubory*“ obsahující veškeré konfigurační soubory, které byly změněny při realizaci GSM sítě (accfile.conf, javascript.conf, regfile.conf, regexroute.conf, subscribers.conf, ybts.conf). Jsou zde také přiloženy upravené soubory usrp_basic.cc a db_bitshark_rx.cc.

V dalším adresáři „*skript*“ se nachází skript k instalaci nejnovější verze od Marcuse Leecha.

Posledním adresářem je „*software*“, ve kterém je uložena verze GNURadia 3.4.2 a program kalibrate_uhd.