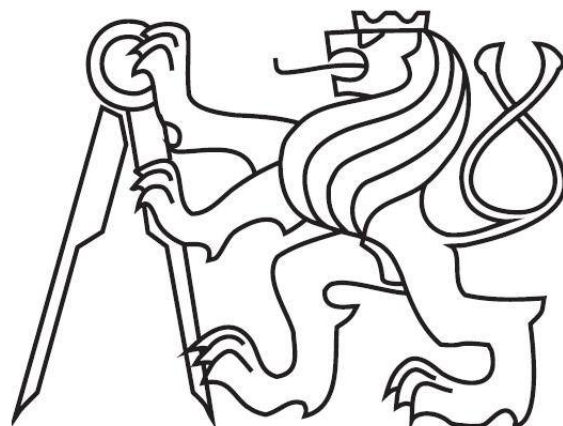


ČESKÉ VYSOKÉ TECHNICKÉ UČENÍ V PRAZE

FAKULTA ELEKTROTECHNICKÁ
KATEDRA TELEKOMUNIKAČNÍ TECHNIKY



DIPLOMOVÁ PRÁCE

Platforma pro testování mobilních sítí

Autor	Bc. Dominik Prekschl
Vedoucí práce	Ing. Pavel Troller, CSc.
Praha	Červen 2016

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra telekomunikační techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Dominik Prekschl**

Studijní program: Komunikace, multimédia a elektronika
Obor: Sítě elektronických komunikací

Název tématu: **Platforma pro testování mobilních sítí**

Pokyny pro vypracování:

Navrhněte systém využívající mobilních měřicích pracovišť postavených na bázi chytrých telefonů platformy Android, která budou vykonávat aktivní i pasivní hovorové testy oproti centrální části platformy, využívající systému Asterisk běžícího pod OS Linux. Evidujte úspěšnost volání, rychlost setupu, kvalitu sestaveného spojení, případně předčasné chybové ukončení spojení atd. Měření nechť probíhá na základě scénáře sestaveného v jednoduchém textovém souboru (např. obsahujícího časy, volaná čísla, délky hovorů, atd). Pro účely prohlížení naměřených dat vytvořte jednoduchou WWW stránku.

Seznam odborné literatury:

- [1] Kreher, R.; Gaenger, K.: *LTE Signaling, Troubleshooting, and Optimization*. Repr. Chichester, West Sussex, U.K: Wiley, 2010. 291 pages. ISBN: 04-706-8900-5.
- [2] 3GPP ETSI TS 136 214 *Physical layer - Measurements*. Dostupné na http://www.etsi.org/deliver/etsi_ts/136200_136299/136214/09.01.00_60/ts_136214v090100p.pdf [on/line].

Vedoucí: Ing. Pavel Troller, CSc.

Platnost zadání: do konce letního semestru 2016/2017



prof. Ing. Boris Šimák, CSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 21. 12. 2015

Anotace

Tato diplomová práce se zabývá možnostmi využití open-source prostředků pro měření výkonnostních parametrů mobilních sítí, zejména pak v operačním systému Linux. Součástí práce je též stručný přehled generací mobilních sítí. S ohledem na skutečnost, že jádrem celé platformy je pobočková ústředna Asterisk, je právě Asterisku věnována jedna celá kapitola této diplomové práce. Pro účely testování bylo sestaveno pět druhů testů. Vyhodnocovanými parametry jsou pak zejména doby sestavení hovorů, kvalita sestaveného spojení a dostupnost mobilní stanice. Naměřené výsledky jsou interpretovány na jednoduché webové stránce, která výsledky jednotlivých testovacích hovorů zobrazuje plně automaticky.

Klíčová slova: Android, Asterisk, GSM, KPI, MOS, PESQ

Annotation

This thesis covers the issue of using open-source tools to measure cellular network's performance within the Linux operating system. The thesis also includes a brief overview of generations of mobile networks. One entire chapter deals with Asterisk private branch exchange since it is a core component of the platform. For the purposes of testing, five different tests have been carried out. The criteria include especially call setup time, connection quality and accessibility. Interpretation of collected data is performed through a simple website and is fully automatic.

Keywords: Android, Asterisk, GSM, KPI, MOS, PESQ

Poděkování

Na tomto místě bych chtěl poděkovat vedoucímu své diplomové práce
Ing. Pavlu Trollerovi, CSc. za velmi užitečnou pomoc a metodické vedení při zpracovávání
této práce.

V Praze dne:

.....

podpis autora práce

Prohlášení autora práce

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne:

.....

podpis autora práce

Obsah

1	Úvod	1
2	Generace mobilních sítí	2
2.1	GSM	2
2.1.1	Stručná historie GSM	2
2.1.2	Základní metody přenosu v GSM síti	3
2.1.3	Struktura GSM sítě	5
2.2	UMTS	6
2.2.1	Základní metody přenosu v UMTS síti	6
2.2.2	Struktura UMTS sítě	6
2.3	LTE	7
2.3.1	Struktura LTE sítě	8
3	Klíčové výkonnostní parametry v GSM	10
4	Kvalita telefonního hovoru a její měření	12
4.1	MOS	12
4.1.1	Vyhodnocení parametru MOS	12
4.2	PESQ	14
4.2.1	Aplikace algoritmu PESQ	16
5	Pobočková ústředna Asterisk	17
5.1	Instalace a spuštění Asterisku	17
5.2	Konfigurační soubory Asterisku	18
5.2.1	Sip.conf	18
5.2.2	extensions.conf	20
5.2.3	rtp.conf	22
5.2.4	iax.conf	22
5.2.5	cdr_custom.conf	25
5.3	Automatické volání z ústředny Asterisk	26
6	Schéma a popis testovací platformy	27
6.1	Vliv IP sítě na naměřená data	29
6.2	Příprava Raspberry Pi	30
6.3	Příprava mobilního telefonu	30

7	Testované parametry	32
7.1	Test dostupnosti - úspěšnosti sestavení hovoru	32
7.2	Test DTMF	33
7.3	Test doby sestavení hovoru	35
7.4	Test SMS zpráv	38
7.5	Test kvality spojení	39
8	Interpretace výsledků	41
8.1	Konfigurace cdr_custom	41
8.2	Struktura webové stránky	42
8.2.1	Převod CDR souborů	44
9	Závěr	45
10	Přílohy	I
10.1	Konfigurační soubory Asterisku	I
10.1.1	cdr_custom.conf	I
10.1.2	extensions.conf	II
10.1.3	iax.conf	III
10.1.4	sip.conf	IV
10.2	Mapování Hangup Cause	IV
10.3	Generátor DTMF	VI
10.4	Vyhodnocovací skript	VII

Seznam obrázků

1	Princip buňkového systému	3
2	Princip sektorizace	4
3	Struktura GSM sítě	5
4	Struktura UMTS sítě	7
5	Struktura LTE sítě	9
6	Průběh mapovací funkce	15
7	Nastavení domácího směrovače	20
8	Schéma testovací platformy	27
9	Testovací přípravek	28
10	Scénář testu doby sestavení hovoru	37
11	Úvodní webová stránka	43
12	Výsledky testu kvality spojení	43

Seznam tabulek

1	ACR stupnice MOS [7]	12
2	Možné kombinace měření MOS [8]	14
3	Traceroute mezi Asterisky	30
4	Výstup algoritmu PESQ	40

Seznam použitých zkratk

ACD	Automatic Call Distribution
ACR	Absolute Category Rating
ADC	Administrative Center
AMR	Adaptive Multi-Rate Compression
AuC	Authentication Center
BSS	Base Station Subsystem
CDR	Call Data Records
CEPT	Conference of Postal and Telecommunications
CLI	Command Line Interface
CN	Core Network
CSSR	Call Setup Success Rate
DCR	Degradation Category Rating
DHCP	Dynamic Host Configuration Protocol
DTMF	Dual-Tone Multi-Frequency
EPC	Evolved Packet Core
ETSI	European Telecommunications Standards Institute
E-UTRA	evolved UMTS Terrestrial Radio Access,
FDMA	Frequency Division Multiple Access
GGSN	Gateway GPRS Support Node
GMSC	Gateway Mobile Switching Centre
GSM	Global System for Mobile Communications
HLR	Home Location Register
HSPA	High Speed Packet Access
HSS	Home Subscriber Server
IAX	Inter-Asterisk eXchange
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ITU	International Telecommunication Union
IVR	Interactive Voice Response
KPI	Key Performance Indicator
LTE	Long Term Evolution
MIMO	Multiple-Input Multiple-Output

MME	Mobility Management Entity
MOS	Mean Opinion Score
MOS-CQE	MOS - Conversational Quality, Estimated
MOS-CQO	MOS - Conversational Quality, Objective
MOS-CQS	MOS - Conversational Quality, Subjective
MOS-LQE	MOS- Listening Quality, Estimated
MOS-LQO	MOS- Listening Quality, Objective
MOS-LQS	MOS- Listening Quality, Subjective
MSC	Mobile Switching Center
NAT	Network Address Translation
NIX	Neutral Internet Exchange
NSS	Network Switching Subsystem
NTP	Network Time Protocol
OFDM	Orthogonal Frequency Division Multiplexing
OMC	Operational and Maintenance Centre
OSS	Operational and Support Subsystem
PBX	Private Branch eXchange
PESQ	Perceptual Evaluation of Speech Quality
P-GW	Packet Gateway
QoS	Quality of Service
RNC	Radio Network Controller
RNS	Radio Network Subsystem
RTP	Real-time Transport Protocol
SAE-GW	System Architecture Evolution Gateway
SDCCH	Stand-alone Dedicated Control Channel
SGSN	Serving GPRS Support Node
S-GW	Serving Gateway
SIM	Ssubscriber Identity Module
SIP	Session Initiation Protocol
SMS	Short Message Service
SSH	Secure Shell
TDMA	Time Division Multiple Access

TCH	Traffic Channel
UDP	User Datagram Protocol
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
UTRAN	UMTS Terrestrial Radio Access Network
VLR	Visitor Location Register
VoIP	Voice over IP
VoLTE	Voice over LTE
WWW	World Wide Web

1 Úvod

V současné době se s narůstajícím počtem aktivně používaných mobilních zařízení stále častěji setkáváme s odmítnutím námi iniciovaného volání z mobilního telefonu. Důvod takového odmítnutí je v zásadě prostý - jen v České republice připadá na 100 obyvatel 130 mobilních telefonů [1], které mezi sebou komunikují pomocí mobilní sítě, a jejich počet neustále narůstá. Tento trend nepochybně souvisí s faktem, že právě mobilní telefony dnes nahrazují většinu platforem pro konzumaci jakéhokoliv obsahu, tedy televizi, denní tisk, rozhlas atd. a jsou používány drtivou většinou obyvatel, a to napříč věkovým i sociálním spektrem. Nahrazují rovněž pevné linky donedávna hojně používané nejen v kancelářích, ale zejména v domácnostech. Mobilní telefony dnes nejsou využívány pouze k volání, ale k mnoha dalším různorodým činnostem. Samotná „telefonní“ funkce, tedy uskutečňování volání, tak již není funkcí jedinou a mnohdy ani funkcí dominantní. Objem celkového provozu generovaného koncovými zařízeními připojenými do mobilní sítě, tedy nejen mobilními telefony, ale také tablety nebo notebooky, stále narůstá. Nárůst počtu mobilních telefonů má nicméně za následek i to, že je nezbytné neustále zvyšovat také výkonnost celé přenosové sítě tak, aby zůstala zachována určitá kvalita služby (QoS) a nedocházelo tak k narůstání nespokojenosti zákazníků, například z důvodu, že se nemohou v určité oblasti dovolat, respektive aby se takové nespokojenosti předcházelo. Lze tedy říci, že výkonnost sítě spolu s parametry QoS mají přímý dopad na celkové postavení mobilního operátora na trhu a tedy i na jeho zisk. Důvody neúspěšnosti volání mohou být různé, od skutečnosti, že v určité oblasti není kvalitní pokrytí signálem mobilního operátora, tedy chybou na rádiovém rozhraní, přes různorodé chyby, které mohou nastat na aktivních prvcích infrastruktury, až po možnou hardwarovou chybu koncového zařízení, na které se účastník pokouší dovolat.

V této práci se tedy zaměřím na klíčové parametry, které klasifikují kvalitu a výkonnost mobilních sítí, a na jejich měření pomocí open-source prostředků, a to spojením platformy Android a softwarové pobočkové ústředny Asterisk. Cílem mé diplomové práce je zjistit, do jaké míry je toto spojení využitelné a kolik parametrů je takto skutečně možné změřit. S ohledem na mně dostupné prostředky nemohu realizovat měření čistě pro současné moderní sítě, omezím se tedy na stále, a pravděpodobně ještě dlouho do budoucna, fungující síť GSM, výsledná platforma však není na generaci mobilní sítě závislá. Existence a fungování GSM sítě je nezbytné pro užívání takových mobilních telefonů, které neumí komunikovat přes modernější infrastrukturu.

2 Generace mobilních sítí

2.1 GSM

2.1.1 Stručná historie GSM

GSM (Global System for Mobile Communication) je mezinárodním standardem pro mobilní komunikaci. Uživatelé se tak mohou do této sítě připojit v zásadě odkudkoliv. Nabízí služby krátkých textových zpráv SMS (Short Message System), faxu nebo hlasové schránky.

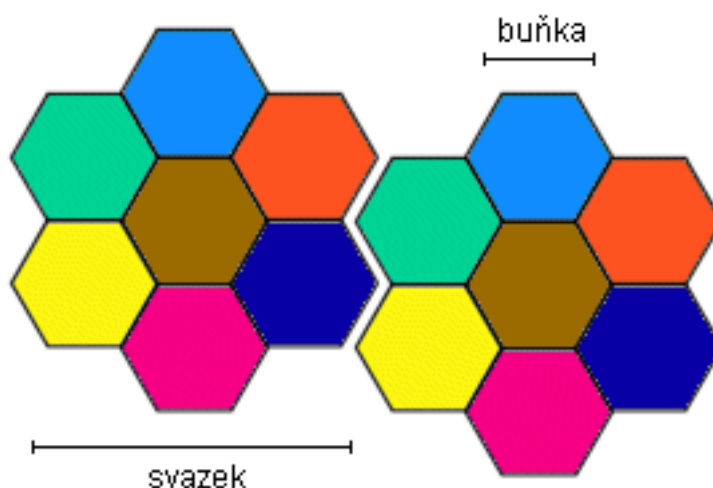
Dalšími vedlejšími službami, které poskytuje, a které přispívají ke zvýšení komfortu uživatele, jsou přeměrování hovorů či zobrazení čísla volajícího, tedy funkce pro dnešního uživatele zcela samozřejmé. GSM využívá několik frekvenčních pásem. Obecně nejběžnějšími frekvenčními pásmy jsou pásma 850 MHz, 900 MHz, 1800 MHz a 1900 MHz.

Historie GSM sahá až do roku 1982 kdy byla Konferencí evropských správ pošt a telekomunikací, CEPT, později ETSI (European Telecommunications Standards Institute), vytvořena pracovní skupina Groupe Spécial Mobile, která navrhla první verzi stejnojmenného standardu. V osmdesátých letech, kdy byl standard vytvořen, však ještě nikdo nepředpokládal, že by mobilní telefony mohly být používány tak, jak tomu dnes je, tedy širokou škálou uživatelů po celém světě. První telefony nebyly, s ohledem na své rozměry a hmotnost, zcela přenosné, z těchto důvodů a též pro svou vysokou spotřebu elektrické energie byly převážně instalovávány do osobních automobilů. Dalo by se říci, že se jednalo o luxusní zboží, i pro obyvatele vyspělých zemí takřka nedostupné, a to jak kvůli vysoké ceně samotných terminálů, tak i z důvodu vysokých cen za uskutečněné hovory. Dalším problémem byla nemožnost realizovat mezistátní hovory z důvodu neexistence roamingových smluv mezi operátory.

Technické základy GSM byly definovány v roce 1987, v roce 1989 pak převzala kontrolu organizace ETSI. V roce 1990 tak vznikla první specifikace GSM, samotný provoz byl zahájen v polovině roku 1991 a o pouhé dva roky později a v roce 1993 bylo v provozu již třicet šest GSM sítí, a to ve dvaadvaceti zemích světa. Systém GSM nezůstal jen evropským standardem, ale rozšířil se po celém světě. Ve všech zemích fungoval, a dodnes funguje, na stejném principu, může však pracovat na různých frekvencích, což ještě v nedávné době způsobovalo, že telefony určené pro evropský trh se nebyly schopné přihlásit do mobilní sítě například amerického operátora a naopak. Dnes už je většina nových telefonů vybavena „multi-band“ anténami, což znamená, že mohou pracovat na různých frekvencích, které jsou po světě pro GSM vyhrazeny. [3]

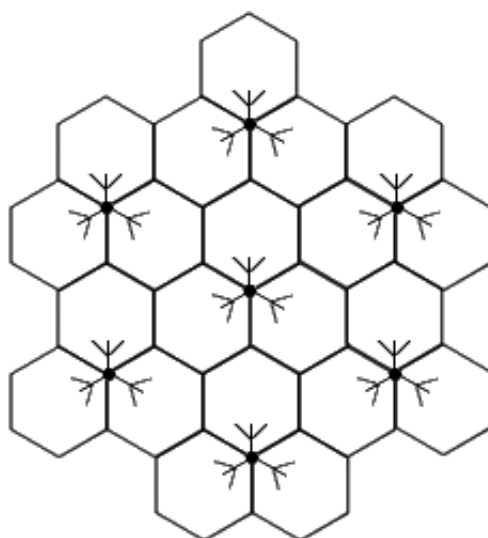
2.1.2 Základní metody přenosu v GSM síti

Celá síť je uspořádána do takzvaného celulárního, nebo také buňkového, systému, který vede k efektivnímu hospodaření s frekvenčním spektrem díky přidělování stejných frekvencí v oblastech, které spolu nesousedí. Každý svazek (cluster) je tvořen sedmi buňkami, z nichž každá představuje určitou geografickou oblast, a v každé z těchto buněk je signál vysílán na jiné frekvenci. Pokud k jednomu svazku připojíme druhý, není potřeba rezervovat další frekvence, ale můžeme použít stejné ze svazku původního. Tímto způsobem tak může být pokryto neomezeně velké území. Princip buňkového systému je znázorněn na následujícím obrázku.



Obrázek 1: Princip buňkového systému

V hustě zastavěných oblastech je pak využíván princip sektorizace, kdy je každý svazek rozdělen na dvacet jedna menších buněk. Pro snížení počtu základnových stanic z jednadvaceti na sedm můžeme každou z nich umístit na rozhraní tří sousedních buněk a použít sektorové antény tak, jak je naznačeno na obrázku níže. Sektorizace je používána za účelem zvětšení kapacity sítě, respektive zvýšení možného počtu současně obsluhovaných účastníků. Je však nutné dodat, že ve skutečnosti nemají buňky vždy ideální tvar šestiúhelníku, ale v podstatě náhodný a nepravidelný. V důsledku této nepravidelnosti tak vznikají interferenční zóny, kde mobilní stanice přijímá signály s různou frekvencí a může tak dojít k rádiovému rušení. V praktických aplikacích pak existují svazky i s jiným počtem buněk než sedm.



Obrázek 2: Princip sektorizace

System GSM využívá dvou různých metod sdílení přenosového média, metodu FDMA a metodu TDMA.

FDMA (Frequency Division Multiple Access) je metoda sdílení jednoho frekvenčního pásma více uživateli, kdy každý uživatel, respektive skupina osmi uživatelů, vysílá a přijímá svá data na specifických frekvencích, které jsou vyhrazeny právě této skupině účastníků a pouze jim jsou vyhrazeny po celou dobu spojení. Celé frekvenční pásmo je tedy rozděleno na konkrétní počet kanálů, které jsou přiřazovány účastníkům.

TDMA (Time Division Multiple Access) je druhou používanou metodou sdílení přenosového média, při které jsou signály odděleny tím, že každý účastník, který v danou chvíli chce využívat daný kanál pro komunikaci, vysílá vždy v pevně daných krátkých časových úsecích či intervalech, které se označují jako časové sloty (Timeslot). Těchto timeslotů je zpravidla osm (0-7) a každý takto vzniklý interval představuje jednoho účastníka. Osm účastníků tedy může ve stejnou chvíli využívat stejnou frekvenci.

Metody jsou používány současně, a to tak, že je frekvenční pásmo, typicky 200 kHz, rozděleno na kanály metodou FDMA a na jednotlivých kanálech se vysílá metodou TDMA. [2]

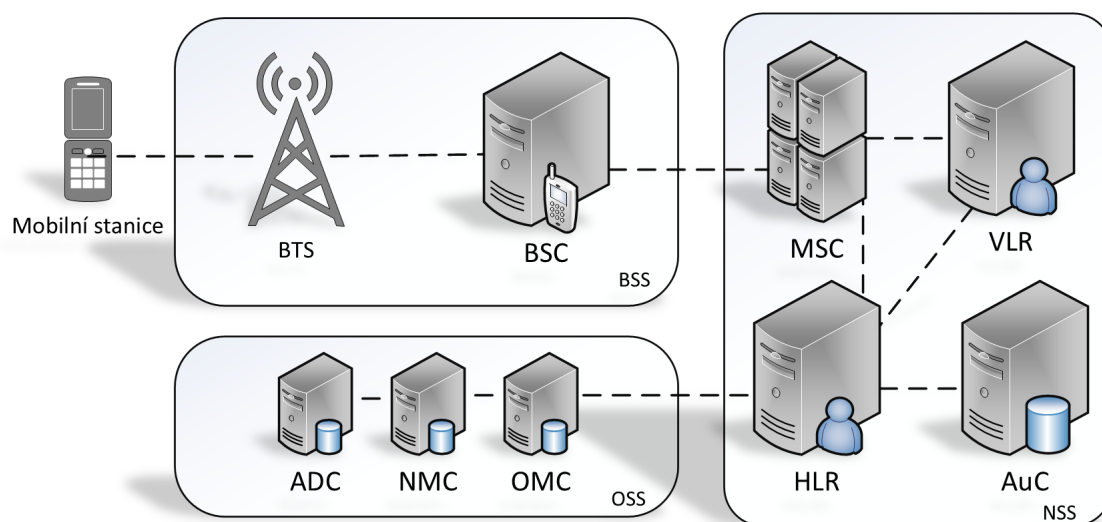
2.1.3 Struktura GSM sítě

Celá síť je rozdělna na tři systémy, a to OSS, NSS a BSS.

OSS OSS (Operational and Support Subsystem) je operační podpůrný subsystém, který je tvořen servisním centrem OMC (Operational and Maintenance Centre), centrem pro řízení sítě NMC (Network Management Centre) a administrativním centrem ADC (Administrative Center). Celý subsystém OSS zajišťuje chod a údržbu GSM sítě, tedy zejména monitoruje mobilní stanice, eviduje porouchané mobilní stanice, konfiguruje síť a spravuje tarifkaci.

NSS NSS (Network Switching Subsystem) je síťový spojovací subsystém, který je tvořen několika částmi, zejména ústřednou MSC (Mobile Switching Center), bránou mobilní ústředny GMSC (Gateway Mobile Switching Center), domovským registrem HLR (Home Location Register), návštěvnickým registrem VLR (Visitor Location Register), centrem autorizace AuC (Authorization Center) a SMS centrem.

BSS BSS (Base Station Subsystem) je subsystém základnových stanic. Je tvořen základnovými stanicemi BTS (Base Transciever Station) a základnovými řídicími jednotkami BSC (Base Station Controller). [3]



Obrázek 3: Struktura GSM sítě

2.2 UMTS

S ohledem na skutečnost, že první GSM sítě vznikaly v době, kdy koncoví účastníci poptávali zejména poskytování hlasových služeb, je GSM síť koncipovaná právě pro přenos hlasu, tedy pro okruhově orientované služby. S rozvojem internetu pak byly tyto sítě pro přenos dat, který je v GSM síti z dnešního pohledu velmi pomalý, dosahuje rychlosti pouze do 30 kb/s, nedostačující. UMTS síť, síť třetí generace, je v podstatě rozšířením sítě GSM, které je zaměřené zejména na přenos dat, tedy paketově orientovanou službu. Oproti GSM síti přinesla síť třetí generace metodu sdílení kanálu CDMA (Code Division Multiple Access), kdy je umožněná komunikace v jednom frekvenčním pásmu (typicky 5 MHz) výrazně většímu počtu uživatelů, než v případě kombinace metod FDMA/TDMA. Hlavní změnou v přístupu k přenosu řečového signálu bylo použití metody AMR pro kódování řeči. Kodeky tohoto typu umožňují měnit výstupní přenosové rychlosti v závislosti na řečové aktivitě mluvčího. Použití detektoru hlasové aktivity pak zajistí to, aby v době, kdy uživatel nehovoří, mobilní stanice přestala vysílat. Tím, že se uživatelé při hovoru v řeči zpravidla střídají, vedou dialog, je ušetřena až polovina přenosové kapacity. Se sítěmi třetí generace pak souvisí zkratka HSPA (High Speed Packet Access, což je rozšíření pro UMTS síť, které je zaměřeno výlučně na přenos dat a poskytuje tak mobilní širokopásmové připojení. [2]

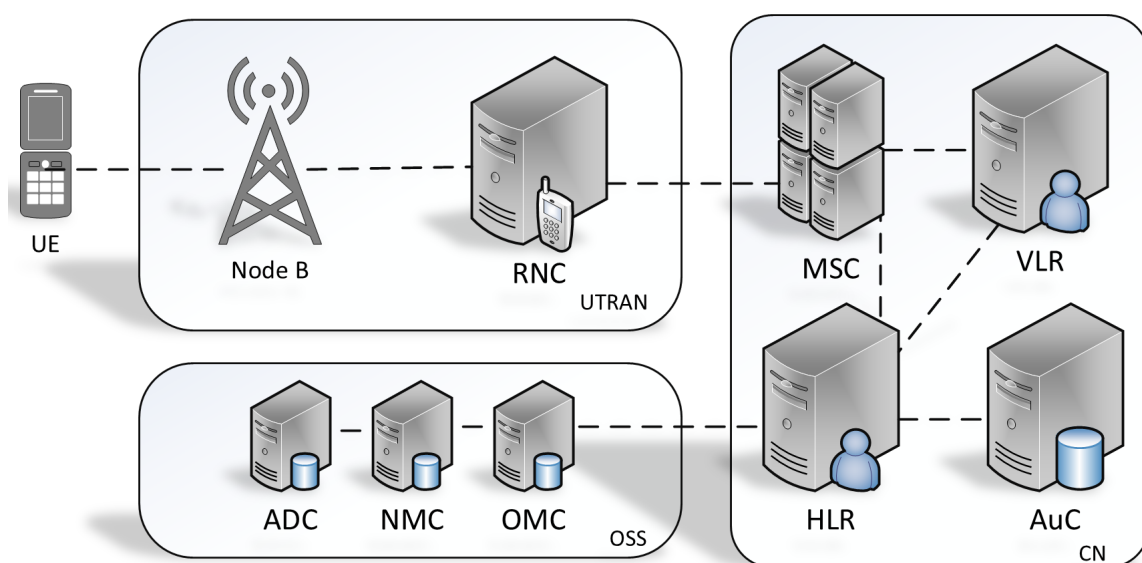
2.2.1 Základní metody přenosu v UMTS síti

UMTS využívá členění na buňky o různých velikostech, zejména pikobuňky, mikrobunčky a makrobunčky. Cílem tohoto členění je zaručení určité kvality služby nezávisle na prostředí. Pikobuňkami jsou označována místa s vysokou koncentrací účastníků, kde tím pádem vznikají vysoké nároky na přenosovou rychlost, takovými místy mohou být například obchodní centra. Mají zpravidla poloměr do padesáti metrů. Naopak makrobunčka označuje oblast, ve které se účastníci rychle pohybují a nároky na přenosovou rychlost jsou tak nižší, tedy například úseky silnic a dálnic. Poloměr těchto buněk může dosahovat desítek kilometrů. Ostatní vlastnosti zůstávají v podstatě totožné s technologií sítě GSM. [2]

2.2.2 Struktura UMTS sítě

Základem UMTS sítě je páteřní síť, tedy CN (Core Network), která je obdobou substému NSS GSM sítě. Tato CN síť se dělí na domény pro zpracování paketově orientovaných služeb a okruhově orientovaných služeb. Podobnost je dána zejména v doméně pro okruhově orientované služby, kam patří telefonní hovory. V té nalezneme telefonní ústředny MSC a GMSC, stejně jako u GSM. V paketové doméně nalezneme uzly SGSN a GGSN,

kteřé, mimo jiné, směrují pakety do vnějších sítí. Další částí UMTS sítě je přístupová síť UTRAN (UMTS Terrestrial Radio Access Network), která je obdobou subsystému BSS v GSM síti a skládá se ze subsystémů rádiových sítí RNS. V UMTS sítích je BTS nahrazena základnovou stanicí Node B, BSC je pak nahrazen řídicí jednotkou rádiové sítě RNC. RNC se tedy stará, stejně jako BSC, o správu rádiových prostředků na území rozděleném na buňky, dále řídí handovery, výkon a kontroluje provoz Node B. Posledními částmi jsou účastnické sady UE (User Equipment), tedy koncová zařízení účastníků. [2]



Obrázek 4: Struktura UMTS sítě

2.3 LTE

Trend spočívající ve zvyšujících se nárocích na přenosové rychlosti datových spojení je hlavním důvodem vzniku další generace mobilních sítí LTE - Long Term Evolution, která bývá často nesprávně označována za síť čtvrté generace. LTE je příprava pro síť LTE - Advance, opravdové 4G síť. LTE je založeno výhradně na protokolu IP, tím je usnadněna komunikace s ostatními pevnými datovými sítěmi, ve kterých je dnes obdobně používán zejména protokol IP. Přepojování okruhů již není podporováno. Lze tak realizovat volání pomocí protokolu IP - VoIP. Je však běžné, že i tehdy, kdy operátor spustí LTE síť, hlasové hovory jsou nadále obsluhovány pomocí starších typů sítí. Volání přes LTE je tak

spíše funkcí doplňkovou, tato síť je zaměřena primárně na datové přenosy, a to s cílem přenos dat zdokonalit a zrychlit, proto je také přínos pro uživatele v tomto směru výrazně vyšší než přínos v podobě zlepšení kvality telefonních hovorů. V LTE jsou používány větší šířky přenosových pásem a více antén v mobilních stanicích (MIMO - Multiple Input, Multiple Output), typicky v režimu 2x2. Další změnou oproti předchůdcům je metoda vícenásobného přístupu, v tomto případě označená OFDMA (Orthogonal Frequency Division Multiple Access). Je tak opět lépe využito celé frekvenční pásmo a tím je dosaženo větších přenosových rychlostí směrem k uživateli. Výhodou je možnost použití na jakékoliv frekvenci, LTE pracuje i mimo licencovaná pásma, typicky od 700 MHz do 2,4 GHz. S ohledem na možné interference s jinými službami je nutný pečlivý výběr vysílacích frekvencí. Na nižších frekvencích lze dosáhnout lepšího šíření signálu a tedy i pokrytí většího území jednou základnovou stanicí. Vyšší frekvence oproti tomu nabízejí větší šířky pásma a tedy vyšší přenosové rychlosti, dosah je pak ale nižší. Maximální hodnota downlinku pak může dosahovat přes 300 Mbit/s, uplink pak do 75 Mbit/s. [25]

2.3.1 Struktura LTE sítě

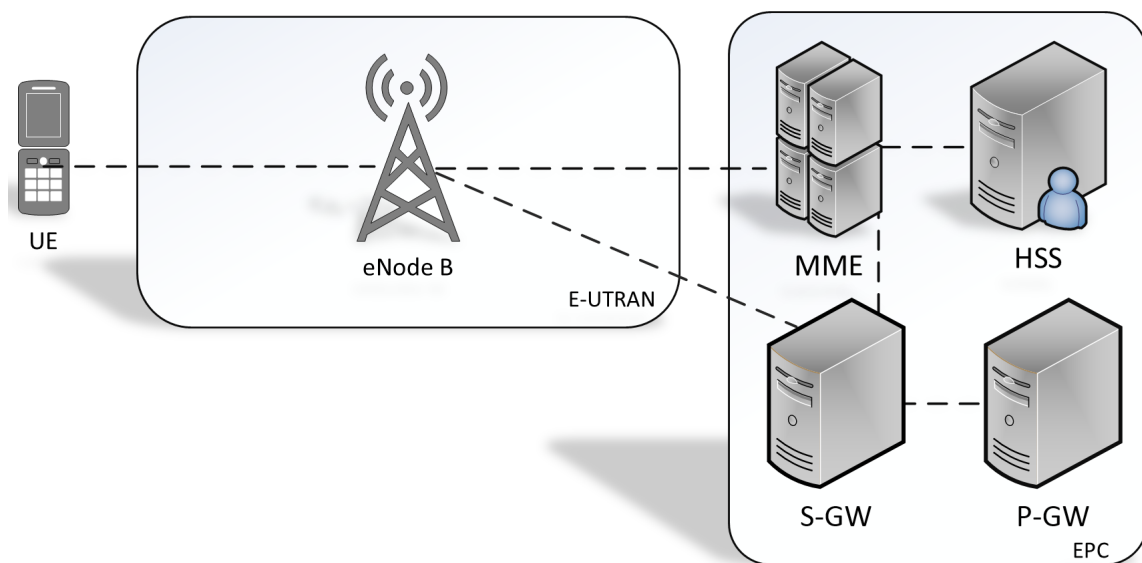
Základem rádiové sítě E-UTRAN je základnová stanice označována eNode B, která současně plní funkci řídicí jednotky rádiové sítě. Dále, mimo jiné, zajišťuje přidělení jednotlivých rádiových kanálů podle priorit a požadované kvality služeb, provádí měření signálu a spolu s hodnotami z mobilní stanice rozhoduje o handoveru nebo například odpovídá za bezpečnost, tedy šifrování a dešifrování IP hlaviček.

Paketová síť EPC (Evolved Packet Core) je pak složena z následujících částí:

- MME (Mobility Management Entity)
 - Hlavní řídicí prvek přes který neprocházejí uživatelská data. Zajišťuje například ověření totožnosti, poskytuje bezpečnostní ochranu komunikace, zajišťuje důvěryhodnost nebo také sleduje pohyb účastníků v dané oblasti a hlásí polohu HSS.
- Domácího uživatelského serveru HSS
 - Server který je spojen se všemi MME v síti, kterým poskytuje kopie uživatelských profilů. V těchto profilech je definováno, ke kterým službám má daný uživatel přístup. Obsahuje i autorizační server, který ověřuje totožnost pomocí klíčů uložených na SIM kartách.

- Brána SAE-GW

- Přes tuto bránu procházejí všechna uživatelská data. Skládá se ze servisní brány S-GW, která řídí tok uživatelských a řídicích dat, a brány paketové sítě P-GW, která plní funkci směrovače mezi EPC a vnějšími paketovými sítěmi.



Obrázek 5: Struktura LTE sítě

3 Klíčové výkonnostní parametry v GSM

Klíčových výkonnostních parametrů existuje velké množství, jejich hodnoty jsou nicméně v zásadě drženy v tajnosti poskytovateli mobilních sítí, a pokud nastane situace, kdy některé z těchto hodnot „uniknou“ na veřejnost, neexistuje záruka jejich integrity, nemusí být věrohodné a nelze z nich proto bezvýhradně vycházet. Jejich získání je pro běžného uživatele v zásadě možné pouze vlastním měřením, které je na samotných operátorech nezávislé.

Jedním z nejdůležitějších parametrů, který lze pro stanovení výkonnosti mobilní sítě a její kvality použít, je parametr CSSR (Call Set-up Success Rate), volně přeloženo jako „úspěšnost volání“. Tento parametr zároveň zásadně ovlivňuje spokojenost zákazníka a tedy i celkový finanční profit operátora. Parametr CSSR je definován jako podíl úspěšných volání k počtu celkových pokusů o volání, kdy k určení CSSR stačí jen velmi krátké hovorové testy, a bývá obvykle udáván v procentech a jeho obvyklá hodnota v mobilních sítích je velmi blízká hodnotě 1, tedy 100%. Za úspěšné volání je pak, mimo jiné, pokládáno i takové volání, kdy se účastník pokouší dovolat jinému účastníkovi, který právě s někým telefonuje, a volající straně se ozve obsazovací tón. V podstatě neexistuje univerzální způsob, jak CSSR parametr určit, mobilní operátoři tak tento parametr mohou měřit různými způsoby.

Aby mohlo být spojení v GSM síti navázáno, musí projít dvěma prvotními na sebe navazujícími procedurami, jež jsou nezbytné pro komunikaci mezi mobilním zařízením a pevnou infrastrukturou mobilní sítě. V první fázi je nutné mobilní stanici vyhradit signalizační kanál SDCCH (Stand-alone Dedicated Control Channel) a následně, ve druhé fázi, provozní kanál TCH (Traffic Channel), který přenáší samotný hovor. Nejčastějším důvodem neúspěšnosti volání je tedy nepřidělení signalizačního nebo následně provozního kanálu. SDCCH je používán nejen při sestavení telefonního hovoru, ale rovněž při registraci účastníka nebo také při přenosu SMS zpráv (rozlišeno pomocí informačního pole ve zprávě Channel Request, kterou vysílá mobilní stanice v případě potřeby přidělení SDCCH). Pokud tedy selže jedna z procedur pro přidělení těchto kanálů, volání je neúspěšné. Pro tyto dva kanály pak existují vlastní KPI - SDCCH BR (SDCCH Blocking Rate) a TCH BR (TCH Blocking Rate).

Mezi další významné výkonnostní parametry patří rychlost sestavení volání. Rychlostí sestavení volání se rozumí doba od okamžiku, kdy mobilní stanice žádá o přidělení komunikačního kanálu, do okamžiku, kdy je volané straně zaslána signalizační zpráva, která označuje začátek hovoru.

Dalšími jsou pak kvalita sestaveného spojení, které je věnována kapitola 4 této práce,

předčasné chybové ukončení spojení nebo úspěšnost handeoverů. Některé z těchto uvedených parametrů není možné změřit bez přístupu k údajům z prvků infrastruktury mobilního operátora. Proto jsou v této práci některé parametry měřeny pouze takovým způsobem, který je možný bez tohoto přístupu dosáhnout.[5, 6]

4 Kvalita telefonního hovoru a její měření

Stanovit kvalitu řečového signálu je obecně poměrně složité, a to zejména z důvodu, že záleží na každém jednotlivci, jak konkrétní hovor zhodnotí. Výsledná hodnota kvality tedy silně závisí na každém posluchači. Tato kvalita může být ovlivněna nejrůznějšími faktory, které na celý telefonní hovor působí, tedy způsobem mluvy řečníka, kvalitou zařízení použitého pro telefonování, tedy kvalitou mikrofону nebo reproduktoru, dále pak vlivem okolního ruchu, použitým kodekem a rovněž samotnou kvalitou přenosového kanálu. Metody pro taková měření jsou stanoveny v doporučeních ITU-T P.10.

4.1 MOS

Nejpoužívanějším parametrem pro posouzení kvality řečového signálu je hodnota parametru MOS, z anglického Mean Opinion Score, což je nejčastěji číselná hodnota ze stupnice 1 - 5, takzvané stupnice poslechové kvality ACR (Absolute Category Rating), jak je uvedeno níže v tabulce 1. Existují však i jiné stupnice, například DCR (Degradation Category Rating), kde hodnotící subjekt hodnotí rozdílnost originálního a degradovaného vzorku, který prošel testovaným kanálem. Z anglického názvu je zřejmé, že výsledná hodnota je průměrem hodnocení od jednotlivých hodnotících subjektů. Parametr MOS není obecně pevně spjat pouze s hodnocením kvality telefonních hovorů, ale lze ho využít například také pro hodnocení celkového audiovizuálního dojmu testované služby a stejně tak pro hodnocení audio nahrávek, které nutně nemusely být degradovány přenosem přes hovorový kanál. [7]

MOS	Kvalita
5	Vynikající
4	Dobrá
3	Průměrná
2	Nízká
1	Špatná

Tabulka 1: ACR stupnice MOS [7]

4.1.1 Vyhodnocení parametru MOS

Pro vyhodnocení parametru MOS existují dvě koncepce testu. První možností je test poslechový, druhou test konverzační. Při poslechovém testu, stejně jako v této práci, je

přehrávána předem známá zvuková nahrávka přes hovorový kanál, kde je buď nahrávána nebo přehrávána posluchačům, a to v závislosti na použité metodě vyhodnocení (viz níže). U konverzačního testu je potřeba přítomnost hodnotícího subjektu na obou koncích hovorového kanálu, přes který hovor probíhá.

Obecně existují tři metody stanovení hodnoty MOS, metoda subjektivní, metoda objektivní a metoda odhadovaná. [7]

Subjektivní metoda

Subjektivní metoda vychází z oznámkování jednotlivých nahrávek velkou skupinou posluchačů tak, aby bylo hodnocení stanoveno s co možná největší přesností. Existuje však několik podmínek, které musí být splněny, aby nedošlo ke zkreslení výsledků. Jednak se testy musí odehrávat ve speciálně navržených laboratořích, které jsou odhlučněné, nebo také pravidlo, že každý posluchač se může zúčastnit testu jen jednou. Tato metoda je označována za nejvíce přesnou, ale také nejnáročnější, a to zejména po stránce finanční.

Objektivní metoda

Druhou metodou, metodou která je zároveň použita v této práci, je metoda objektivní. Ke stanovení hodnoty MOS není nutná přítomnost posluchačů a výsledná hodnota je stanovena pomocí důkladně navrženého algoritmu tak, aby co nejvíce odpovídala hodnotě získané subjektivní metodou. Výsledky stanovené pomocí této metody však nemohou být tak přesné, jako ty, které jsou získané metodou subjektivní. Výhodou metody objektivní oproti subjektivní je pak možnost sledování výsledků v reálném čase.

Odhadovaná metoda

Odhadovaná metoda ke stanovení parametru MOS přistupuje zcela jinak než předchozí dvě uvedené metody. Bývá využito pouze parametrického popisu daného systému spolu s kvalitativními parametry. Tato metoda pak nezohledňuje dynamické jevy, které jsou unikátní pro každý sestavený hovor, a je využívána zejména při návrzích nových systémů pro získání rámcové představy o kvalitě hovorů, které budou pomocí tohoto systému realizovány.

[8]

Po zkombinování dvou základních koncepcí testu a tří metod získání parametru MOS dostáváme šest možných kombinací, ty jsou uvedeny v následující tabulce.

Test/Metoda	Pasivní	Aktivní
Subjektivní	MOS-LQS	MOS-CQS
Objektivní	MOS-LQO	MOS-CQO
Odhadovaná	MOS-LQE	MOS-CQE

Tabulka 2: Možné kombinace měření MOS [8]

Tato práce se zaměřuje na kombinaci MOS-LQO kam patří i použitý algoritmus PESQ.

4.2 PESQ

Algoritmus PESQ (Perceptual Evaluation of Speech Quality) se řadí mezi objektivní metody intrusivní, tedy takové metody, které pro stanovení parametru MOS potřebují znát jak originální tak i degradovaný signál. Opakem intrusivních metod jsou metody neintrusivní, které parametr MOS stanovují pouze z degradovaného signálu.

Algoritmus je podle [9] výsledkem několikaletého vývoje a je primárně určený pro vyhodnocování kvality hovoru v základním hovorovém pásmu šířky 3,1 kHz. Je také nutné zdůraznit, že tento algoritmus je vhodný pouze pro pasivní, respektive jednosměrné testování. To znamená, že parametry jako ztráta hlasitosti během hovoru, ozvěna či zpoždění tímto algoritmem nejsou žádným způsobem zohledněny. Vstupem do tohoto algoritmu jsou dva signály, tedy signál originální a signál degradovaný, který prošel komunikačním systémem. Výsledkem je odhad hodnocení kvality, které by bylo uděleno hodnotícím subjektem při subjektivních poslechových testech.

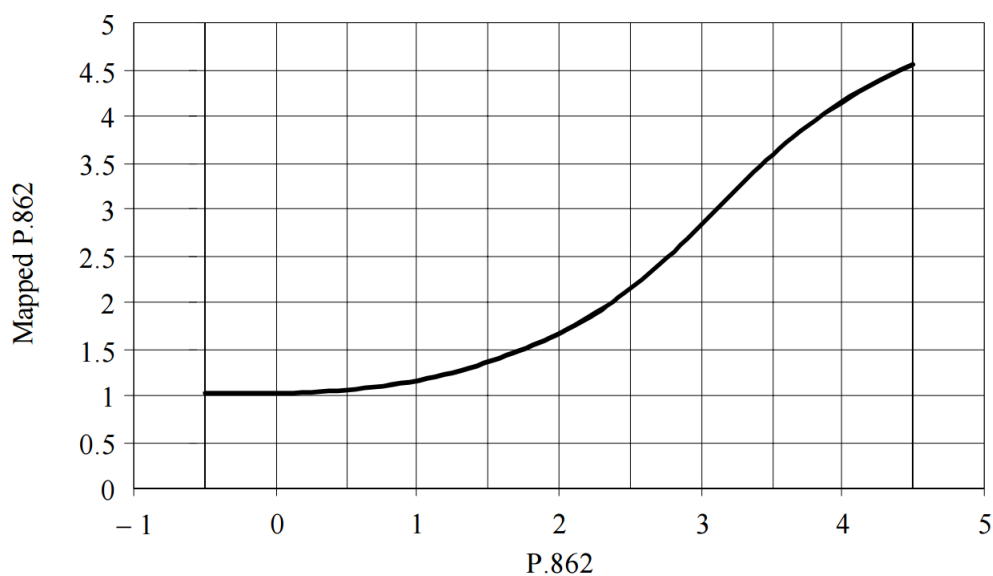
V prvním kroku algoritmus počítá časové zpoždění mezi originálním a degradovaným signálem. Signály jsou v tomto kroku rozděleny na bloky a výsledkem jsou pak body, ve kterých tyto bloky přesně začínají a končí. Odpovídající bloky z obou signálů se pak časově zarovnají. V dalším kroku algoritmus porovnává vstupní signály, základem tohoto porovnání je převedení obou signálů na tvar, který je analogický k psychofyzické reprezentaci zvukových signálů lidského sluchového aparátu ve smyslu rozdílnosti vnímání hlasitosti s měnící se frekvencí zvukového signálu. Pomocí této transformace se algoritmus snaží zachytit důležité veličiny, které mají přímý dopad na vnímanou kvalitu hlasového signálu. Tyto veličiny jsou pak porovnávány u originálního i degradovaného signálu s cílem odhadnout změnu kvality po průchodu komunikačním řetězcem. Výsledná hodnotící stup-

nice parametru MOS získaného algoritmem PESQ má odlišný rozsah oproti stupnici ACR. Konkrétně může výsledné skóre nabývat hodnot -0,5 - 4,5.

Aby bylo možné hodnoty získané pomocí PESQ porovnávat s hodnotou MOS a tedy také přímo porovnávat výslednou kvalitu například s objektivními metodami, vydala ITU-T doporučení P.862.1 [10], které definuje mapovací funkci pro převod hodnoty získané pomocí PESQ na hodnotu standardní ACR stupnice.

$$y = 0,999 \frac{4,999 - 0,999}{1 + e^{-1,4945*x+4,6607}}$$

Tuto rovnici jsem tedy musel zohlednit při zpracování výsledků, abych mohl kvalitu hovoru vyhodnocovat pomocí parametru MOS.



Obrázek 6: Průběh mapovací funkce

4.2.1 Aplikace algoritmu PESQ

Spolu s doporučením, které se týká popisu samotného algoritmu, vydala ITU ještě další doporučení, které popisuje praktické poznámky k použití a implementaci tohoto algoritmu. Je tedy vhodné zmínit vybrané pasáže. Pro získání přesných výsledků je například vhodné použít jako referenci signály o délce 8-12 sekund, nicméně lze aplikovat signály až o délce třiceti vteřin. V zásadě tak lze pro referenci použít signály o délce 8-30 vteřin včetně ticha, které se může vyskytnout před nebo za samotnou řečovou stopou. Řečová aktivita by u těchto signálů měla dosahovat hodnot mezi 40-80 %. Zároveň by se v referenčních signálech měly vyskytovat přirozené pauzy mezi jednotlivými slovy či mezi jednotlivými větami tak, jako se vyskytují v běžné řeči.

Spolu se staženým archivem, ve kterém se nachází zdrojové soubory algoritmu, se v archivu nachází i sada testovacích zvukových souborů, které slouží k ověření správné funkčnosti algoritmu. U každého vzorku je dostupný originální i degradovaný signál. K těmto souborům jsou tedy dostupné i výsledné hodnoty parametru PESQ, které lze porovnat s výsledky, které vyprodukuje algoritmus na konkrétním systému, kde je implementovaný. K otestování správné implementace je zároveň dostupný skript, který načte všechny dostupné zvukové soubory, které jsou součástí archivu, a pro všechny vypočte hodnoty parametru PESQ, které lze porovnat s referenčními výsledky. V této práci jsem pro testování použil řečový signál, který byl součástí archivu.

Mnoho poznámek v tomto doporučení pak hovoří o hlasitosti řečové stopy, která bude přenášena komunikačním kanálem. Hlasitost by se měla pohybovat okolo hodnoty -30 dBov, aby bylo vyloučené případné přebuzení vstupu zařízení, které hovory nahrává. Samotná výkonová úroveň řečové aktivity je však v rámci výpočtu hodnoty PESQ normalizována algoritmem a z toho důvodu tak nemůže být algoritmus použit pro vyhodnocení změny úrovně hlasitosti vzniklé během přenosu signálu. [26]

5 Pobočková ústředna Asterisk

Asterisk je jedním z mála open-source softwarů, které dokáží plně nahradit telefonní ústřednu všech typů, a to jak analogovou, tak i digitální či plně paketovou. Tvůrci Asterisku, americká společnost Digium, Inc., ho označují jako softwarovou pobočkovou ústřednu - PBX (Private Branch eXchange), avšak samotný software dokáže zastat nespočet dalších funkcí, které pobočková ústředna v zásadě nepotřebuje. Těmito funkcemi je například IVR (Interactive Voice Response), interaktivní hlasový průvodce, který může být ovládán tónovou volbou DTMF nebo hlasem, dále pak automatická obsluha hovorů, hlasová schránka, Gateway, tedy převod mezi odlišnými protokoly nebo možnost konferenčních hovorů. Systém Asterisk tak obecně může sloužit pro základ menší až střední telefonní sítě, velikost sítě je však omezena hardwarovými prostředky, které jsou Asterisku vyhrazeny. Hardwarové prostředky jsou tedy v zásadě jediným omezením velikosti sítě. V této práci je hardware zastoupený starším laptopem Acer s procesorem Intel Core 2 Duo T5870 o frekvenci 2 GHz a 3 GB paměti RAM, což je pro plynulý a bezproblémový provoz z hlediska hardwarové stránky zcela dostačující.

Společnost Digium, Inc. (a rovněž další výrobci) nabízí na trhu také specializovaný hardware pro tuto platformu ve formě přídatných karet a modulů, kterými můžeme například zajistit komunikaci pro první řád plesinchronní digitální hierarchie E1 nebo pro systém ISDN. Tento specializovaný hardware se typicky připojuje na standardní PCI sběrnici. Po softwarové stránce je Asterisk rozdělen na jádro a několik konfiguračních modulů. Tyto moduly, pokud jsou dostupné, ústředna načítá a připojuje ihned při spuštění. V zásadě se jedná o textové soubory se specifickým názvem a příponou. Pokud se administrátor rozhodne některý z modulů překonfigurovat při běhu ústředny, musí pak tento modul manuálně znovu načíst pomocí konzole Asterisku specifickým příkazem. Tyto příkazy budou vysvětleny v dalších částech této práce. Konfigurační soubory k jednotlivým modulům jsou standardně uloženy ve složce `/etc/asterisk` souborového systému unixových operačních systémů. [11]

5.1 Instalace a spuštění Asterisku

Pro instalaci Asterisku jsem využil standardní příkaz pro instalaci v systému Ubuntu. U tohoto a rovněž i u následujících příkazů je nutné, aby byl uživatel přihlášený jako administrátor systému, tedy `root`.

```
root@ubuntu:~# apt-get install asterisk
```

V mém případě a době, kdy tato instalace probíhala, byla dostupná verze 11.7.0.
Po instalaci lze softwarovou ústřednu spustit příkazem:

```
root@ubuntu:~# asterisk
```

Pokud se ústředna po instalaci automaticky spustí nebo pokud se chce administrátor k jejímu konfiguračnímu prostředí připojit při jejím běhu, je nutné použít příkaz:

```
root@ubuntu:~# asterisk -r
```

nebo:

```
root@ubuntu:~# rasterisk
```

Užitečnou funkcí je pak použití následujícího parametru:

```
root@ubuntu:~# rasterisk -vvvv
```

Počet hlásek „v“ (z anglického slova „verbose“) pak určuje, do jaké míry bude konzole Asterisku podrobná, a kolik bude vypisovat informačních hlásek, respektive jak moc detailní tyto hlášky budou a kolik podrobností se administrátor při sledování konzole dozví.

Těmito základními příkazy tedy lze nainstalovat a spustit pobočkovou ústřednu Asterisk v čisté podobě. Pro její funkčnost je samozřejmě nutné nastavit několik parametrů.

5.2 Konfigurační soubory Asterisku

5.2.1 Sip.conf

Jak již bylo uvedeno výše, v úvodu kapitoly 5, konfigurační soubory jsou standardně dostupné v adresáři `/etc/asterisk`. Všechny tyto soubory mají čistě textovou podobu a jejich modifikace je možná libovolným textovým editorem přímo ze systémového terminálu. Po základní instalaci Asterisku je jich k dispozici necelých sto. Tím je také usnadněná orientace, to znamená, že administrátor nemusí hledat v jednom sáhodlouhém konfiguračním souboru, místo toho si může otevřít podstatně kratší soubor, ve kterém najde relevantní nastavení ke chtěnému modulu, funkci nebo protokolu.

V této práci bylo využito těchto konfiguračních souborů: `asterisk.conf`, `cdr_custom.conf`, `extensions.conf`, `iax.conf`, `rtp.conf` a `sip.conf`. Pomocí dalších souborů a jejich

konfigurací by pak zajisté bylo možné celou testovací platformu vylepšit. Jedním z prvních souborů, které je třeba editovat, je soubor `sip.conf`. Celý příkaz pro editaci tedy vypadá takto.

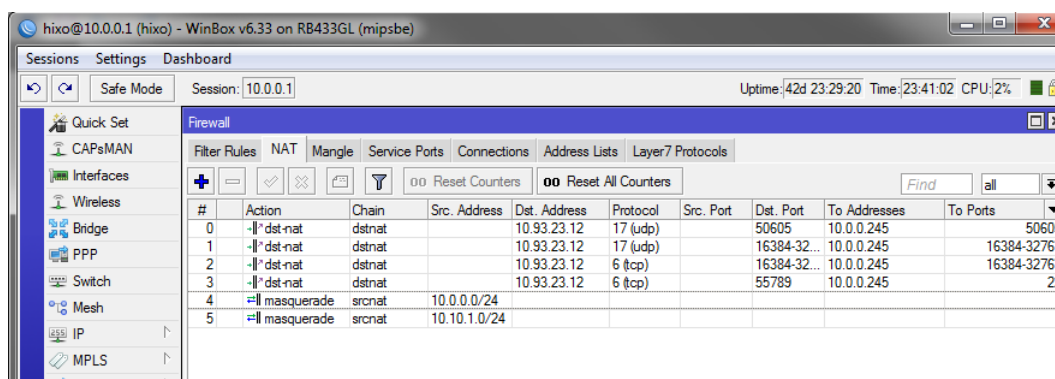
```
root@ubuntu:/etc/asterisk# nano sip.conf
```

Pro Asterisk je typické dělení konfiguračních souborů do částí oddělených jménem v hranatých závorkách. První takovou částí v souboru `sip.conf` je sekce `[general]`. V této sekci je určena veřejná IP adresa, na které bude ústředna dostupná z internetu, a port, na kterém bude Asterisk očekávat signalizační zprávy protokolu SIP. Z bezpečnostních důvodů jsem zvolil jiný, než standartní port, který je pro tento protokol vyhrazený. Dalším parametrem je definování adresního prostoru v rámci lokální sítě, ve které je ústředna umístěna, pokud není přímo připojená do internetu. Pokud je ústředna umístěna v lokální síti, kde je pro přístup k internetu využito překladu adres NAT, je nutné tuto skutečnost Asterisku sdělit. K tomu slouží parametr `nat`. Posledním parametrem je v mém případě bezpečnostní pravidlo `allowguest`, které zakazuje zpracování anonymních volání, která by z veřejné sítě mohla k ústředně dorazit. V mém případě pak sekce `general` ze souboru `sip.conf` vypadá takto.

```
[general]
allowguest=no
bindport=50605
externip=78.108.96.251
localnet=10.0.0.0/255.255.255.0
nat=force_rport,commedia
```

Nedílnou součástí nastavení dostupnosti Asterisku, který je umístěn v lokální síti, je pak přidání výjimek do překladové tabulky směrovače, za kterým je ústředna připojená. Typicky jde o funkci označovanou jako překlad portů. Veškerý provoz určený ústředně jsem tedy z veřejné adresy přeložil na lokální IP adresu Asterisku a na předem definované porty. Z obrázku níže je patrné, že IP adresa, na kterou přichází provoz ze sítě internet, nesouhlasí s IP adresou uvedenou v předchozím konfiguračním souboru. Veřejná IP adresa 78.108.96.251 je překládána až na hraničním směrovači mého poskytovatele internetového připojení metodou „full-cone NAT“. IP adresa 10.93.23.12 je tedy lokální adresou v rámci sítě poskytovatele internetového připojení, která je při opuštění této sítě přeložena na veřejnou IP adresu. Pro možnost vzdálené konfigurace jsem také přemapoval provoz na

veřejnou IP adresu se zdrojovým portem 55789 na lokální IP adresu Asterisku a port 22, tedy na port používaný protokolem SSH. Zcela odlišné číslo zdrojového portu je opět voleno z bezpečnostních důvodů. Při ponechání standardního portu pro protokol SSH by k ústředně mohly přicházet až tisíce žádostí o připojení denně ze serverů, které skenují veřejné IP adresy na Internetu s jediným cílem, a to škodit. Druhou možností vzdálené konfigurace je pak připojení do virtuální privátní sítě.



Obrázek 7: Nastavení domácího směrovače

V další části souboru `sip.conf` je možné definovat jednotlivé účastníky, kteří budou přes ústřednu volat pomocí protokolu SIP. Pro prvotní testovací účely jsem funkčnost Asterisku ověřoval právě pomocí VoIP volání, pro finální podobu celé testovací platformy je pak toto nastavení nepodstatné a podrobnosti k definování nových účastníků nemají větší význam. Celý konfigurační soubor `sip.conf` bude stručně komentován v příloze této práce. [14]

5.2.2 extensions.conf

Soubor `extensions.conf` je v Asterisku zásadně označován pojmem „dialplán“. Celý dialplán je rozdělen na takzvané kontexty, názvy v hranatých závorkách. Toto členění je důležité při vytváření nových komunikačních kanálů v ostatních konfiguračních souborech (`sip.conf`, `iax.conf`...), a to hlavně z důvodu bezpečnosti. Například každému SIP klientovi můžeme přiřadit kontext, ke kterému má možnost přistupovat. Pokud budeme chtít tomuto klientovi například zakázat volání mimo ústřednu Asterisk, přiřadíme mu takový kontext, ve kterém tento typ volání jednoduše nebude definován.

V každém kontextu pak můžeme definovat několik poboček.

Vytváření nových poboček můžeme označit za psaní programového kódu ve speciálním jazyce ústředny Asterisk. Jednotlivé pobočky jsou na sobě nezávislé, pokud tato vlastnost není přímo vyžadována, jako tomu je například u IVR. Syntaxe je vždy velmi podobná, každá nová pobočka začíná slovem **exten** a dále znaky =>. Poté následuje jméno, respektive číslo pobočky, priorita a aplikace, která bude spuštěna. Pro přehlednost vysvětlím výše popsané na jednoduchém příkladu zápisu nové pobočky.

```
exten => 6878,1,Answer()
exten => 6878,n,Playback(/var/lib/asterisk/sounds/en/hello-world)
exten => 6878,n,Hangup()§
```

Uvedený kód zajistí následující to, že pokud účastník vytočí číslo 6678, ústředna toto volání automaticky přijme, účastníkovi přehraje zvukový soubor z uvedené cesty a poté zavěsí. Automatické přijmutí hovoru je zajištěno aplikací **Answer**, přehrání souboru aplikací **Playback** a zavěšení pak aplikací **Hangup**. Celý proces pro jedno konkrétní volené číslo musí vždy začínat řetězcem **exten => X**, kde **X** reprezentuje ono volené číslo, tím je tak zajištěna izolace od ostatních poboček. Co se týče zmíněné priority, ta je v uvedeném dialplanu reprezentována číslicí 1 a dále pak znaky **n**. Prioritou je tedy definováno pořadí aplikací, které budou během hovoru spuštěny. Každá nová pobočka by měla začínat prioritou 1, poté mohou číslice po jedné růst spolu s narůstajícím počtem řádků, nebo můžeme prioritu po jedničce nahradit písmenem **n**. Taková priorita se nazývá nečíslovaná a vznikla z důvodu zjednodušení vytváření nových poboček, přesněji řečeno ke zjednodušení budoucích modifikací. Pokud by bylo použito číslovaných priorit například u dvaceti řádkové pobočky a administrátor Asterisku by chtěl například přidat nový řádek po prioritě číslo 10, musel by všechny následující priority adekvátně a hlavně ručně modifikovat. Písmeno **n** místo číslice jednoduše znamená „další“ z anglického slova „next“ a počítá tak s použitím předchozí hodnoty priority, ke které je přičtena jednička. Tímto způsobem je tak významně zjednodušena práce s prioritami. Dalším zjednodušením práce s dialplánem je použití operátoru **same =>**. V ukázce kódu níže vidíme, že volené číslo „6678“ je psané na každém řádku. Operátor **same =>** je možné použít na řádku, před kterým je použit operátor **exten =>** a není pak dále potřeba psát název pobočky, ale pouze prioritu a aplikaci, viz následující příklad zjednodušení předchozího kódu.

```
exten => 6878,1,Answer()
same => n,Playback(/var/lib/asterisk/sounds/en/hello-world)
same => n,Hangup()
```

Tento kód vykonává tutéž funkci, jako kód předchozí. Operátor `exten` tak usnadňuje kopírování chování pro jednotlivou pobočku na pobočky ostatní, které vykonávají stejnou funkci, protože pak již není potřeba přepisovat názvy poboček.

V praxi pak definice poboček mohou vypadat mnohem složitěji, alespoň na první pohled. Asterisk je v tomto směru velmi komplexní. Zkušený administrátor tak při správné formulaci požadavků na jednotlivé pobočky může i velmi složitý požadavek formulovat na jediném řádku v dialplánu. Funkce dialplánu, které jsem použil v této práci, budou popsány v následujících kapitolách. Popis veškerých funkcí by mohl vydat na samostatnou práci a z hlediska zadání a cíle této diplomové práce není nutné je rozebírat do detailu. [15]

5.2.3 `rtp.conf`

V tomto konfiguračním souboru jsem přenastavil hodnoty počátečního a koncového portu pro přenos RTP streamů, a to na následující hodnoty.

```
rtpstart=16384  
rtpend=32767
```

Toto nastavení bylo důležité zejména v samotných počátcích testování, kdy jsem volal přes IP síť a bylo potřeba sjednotit rozsah těchto portů v ústředně a klientské aplikaci pro systém Android tak, aby nedocházelo k chybám při přenosu hlasu. Při tomto testování jsem, mimo jiné, zjistil, že můj mobilní operátor tyto RTP streamy ve své síti blokuje a nemám tak možnost volat pomocí protokolu VoIP přes „mobilní data“.

Ve finální podobě celé testovací platformy není nutné tyto porty specifikovat, protože v testovaném komunikačním řetězci není RTP stream zastoupen a je nahrazen protokolem IAX. Při rozšíření platformy na IP síť je však velmi důležité toto nastavení zohlednit. [16]

5.2.4 `iax.conf`

IAX (Inter-Asterisk eXchange) je protokol pro sestavení, udržování a ukončení multi-mediálního toku skrze IP síť. Primárně byl tedy zaměřen na VoIP volání, lze ho ale využít například i pro streamování videa či jiného multimediálního obsahu. Díky tomu, že protokol pro přenos využívá pouze jeden UDP stream na předem definovaném portu, typicky 4569, je vhodný pro použití v sítích, ve kterých je používán překlad adres (NAT) a není tedy nutná zvláštní konfigurace firewallů, oproti kombinaci SIP/RTP, kdy taková zvláštní

konfigurace nutná je. Nakonec se protokol IAX vyprofiloval na řešení, které je určeno pro přímé propojení dvou ústředen Asterisk. Protokol má však určité nevýhody. Jednou z nich je fakt, že protokol pro přenos používá veřejně známý port a je tak možným terčem útoku typu DoS (Denial-of-Service). [13]

Konfigurace `iax.conf`

```
[general]
```

```
[k332]
```

```
type=friend
username=tester
secret=*****
auth=md5
host=147.32.*.*
context=public
qualify=yes
qualifysmoothing = yes
jitterbuffer=no
trunk=no
connectedline=yes
disallow=all
allow=g722
allow=alaw
```

- `[k332]` - název IAX trunku
- `type user = friend`
 - definuje typ uživatele; „friend“ dovoluje uživateli sestavovat i přijímat hovory, na rozdíl od typu „peer“ - je dovoleno pouze přijímání hovorů, nebo typu „user“ který hovory může pouze sestavovat
- `username=tester`
 - jméno, kterým se ústředna registruje
- `secret`

- heslo
- `auth=md5`
 - volba autentizačního protokolu
- `host`
 - IP adresa protějšší ústředny
- `context=public`
 - pro jakou sekci číslovacího plánu bude použitý IAX trunk
- `qualify=yes`
 - obecně tento parametr slouží pro definici ověřování dostupnosti protějšší strany; pokud je nastaveno „yes“, PING zprávy jsou odesílány periodicky a pokud odpověď nedorazí do dvou vteřin, je protějšší strana označena za nedostupnou
- `qualifysmoothing= yes`
 - pokud je zvoleno „yes“, Asterisk průměruje dva poslední časy odezvy, která je monitorována parametrem „qualify“, toto nastavení tak snižuje pravděpodobnost, že protějšší strana bude označena jako „lagged“, tedy dostupná s velmi vysokou odezvou, zejména pak při nestabilitě přenosové sítě
- `jitterbuffer = no`
 - jitterbufer slouží pro kompenzaci jitteru vzniklého při přenosu hovoru, v tomto případě je buffer vypnutý, protože by mohl ovlivňovat naměřené hodnoty při testování sítě
- `trunk=no`
 - vypnutí zřetězení RTP hlaviček
- `connectedline=yes`
 - povolení přenosu signalizačních zpráv přes IAX trunk

- `disallow=all, allow=g722, allow=alaw`
 - definice kodeků pro hovory, první volba zakazuje použití všech kodeků, druhá a třetí možnost pak vymezuje použití pouze kodeků g722 a alaw

5.2.5 `cdr_custom.conf`

CDR (Call Data Records) je soubor, do kterého Asterisk přepisuje různé informace ke každému hovoru, který byl ústřednou obslužen, či se o jeho obslužení ústředna snažila, pokud tato funkce není pro pobočku explicitně zakázána. Typicky pak každý hovor reprezentuje jeden řádek hodnot, které jsou v základní konfiguraci odděleny čárkami a vloženy do uvozovek. Tento soubor je svou strukturou vhodný pro následné zpracování různými technikami a způsoby a slouží jako podklad pro takzvaný billing, tedy k vyhodnocení kolik minut jednotliví účastníci protelefonovali a jakou částku má operátor jakému uživateli vyfakturovat.

Mezi hlavní údaje, které Asterisk dokáže zjišťovat a měřit patří následující údaje:

- `Accountcode` - číslo telefonního účtu
- `Src` - označení volajícího
- `Dst` - označení volaného
- `Channel` - který komunikační kanál byl pro hovor použit
- `Lastapp` - poslední aplikace použita v dialplanu
- `Start` - začátek hovoru (datum a čas)
- `Answer` - čas, kdy byl hovor přijat
- `End` - ukončení hovoru
- `Duration` - celková délka hovoru od okamžiku `Start` do `End`
- `Billsec` - délka hovoru od okamžiku `Answer` do `End`; účtované vteřiny
- `Disposition` - výsledný stav volání - přijato, bez odezvy, obsazeno
- `Userfield` - uživatelské pole, které lze definovat v dialplanu, bude vysvětleno v kapitole sedm u jednotlivých testů
- `Hangupcause` - výsledný stav volání, ale více podrobný než položka `Disposition`, obdoba standardu ITU Q.931, význam jednotlivých polí je součástí přílohy práce

5.3 Automatické volání z ústředny Asterisk

Pro kompletnost všech využitých funkcí je potřeba zmínit ještě jednu funkci, kterou ústředna Asterisk nabízí. Pokud je z nějakého důvodu žádáno, aby ústředna sestavila hovor k vybranému uživateli, je možné využít příkazu `originate` přímo z CLI Asterisku. V zásadě lze rozlišit dvě možnosti použití. Jednou z těchto možností je způsob, kdy se sestaví hovor pro konkrétní hovorový kanál a vykoná se nebo se spustí konkrétní aplikace. Druhou variantou je pak sestavení hovoru pro hovorový kanál a konkrétní pobočku z dialplánu, tedy spíše spuštění posloupnosti aplikací.

Pro sestavení hovoru ke konkrétnímu uživateli lze využít ještě jednoho dalšího způsobu. Pokud je do specifického adresáře v souborovém systému umístěn textový soubor se správnou syntaxí, Asterisk ho okamžitě zaregistruje a dle instrukcí uskuteční hovor. Výchozí cesta, kam umisťovat tyto soubory je `/var/spool/asterisk/outgoing`. Syntaxe souboru je pak následující.

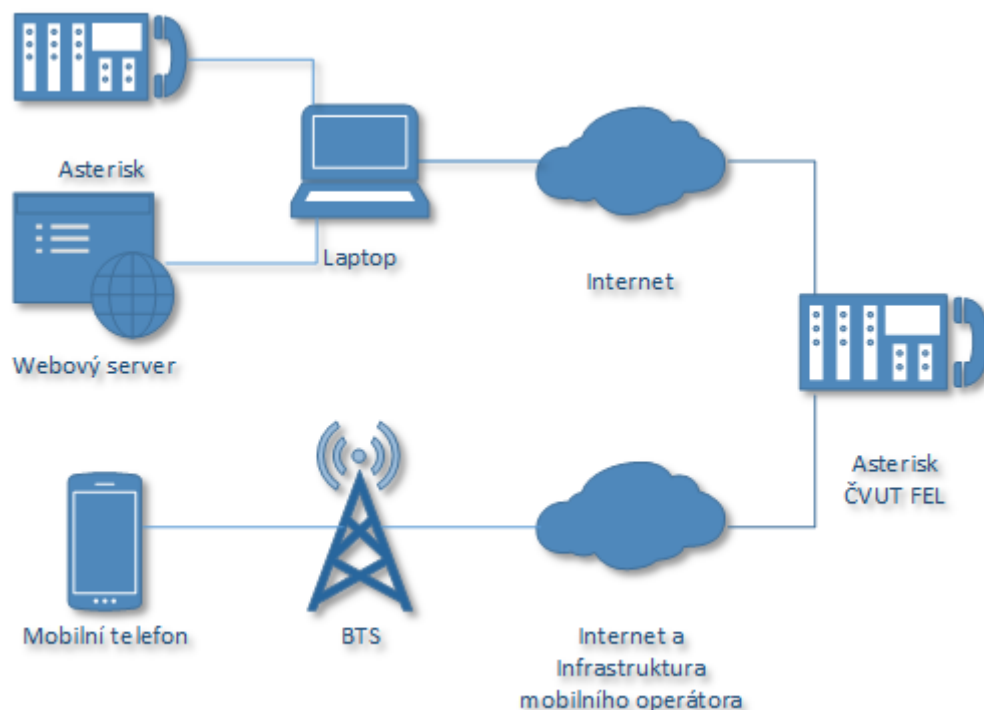
```
Channel: IAX2/k332/0797990104
Application: Playback
Data: hello-world
```

Tyto tři řádky pak zajistí, že se z ústředny sestaví hovor konkrétním směrem, ke konkrétnímu uživateli a použije se aplikace `Playback`, pomocí které se přehraje soubor `hello-world`. Po dokončení hovoru je soubor, který hovor vyvolal, automaticky smazán.

Pro automatizované měření jsem však hledal způsob, jakým hovory naplánovat tak, aby bylo z ústředny voláno například každých 10 minut. Využil jsem programu `Crone`, který je dostupný v linuxové distribuci Ubuntu již po čisté instalaci. Tento program plní funkci plánovače, to znamená, že v uživatelem definované doby vykoná určitý příkaz nebo spustí skript. Připravený soubor, který zahájí sestavení hovoru je tak možné umístit do libovolné složky, následně pomocí příkazu `chown` změnit vlastníka tohoto souboru, konkrétně na uživatele „asterisk“, a pomocí příkazu `crontab -e` založit nový úkol v plánovači, který bude tento soubor v pravidelných intervalech kopírovat do výchozí cesty pro odchozí hovory. Toto je tak jeden z možných existujících způsobů, jak lze zajistit automatické sestavování telefonních hovorů v předem definovaných časových okamžicích. [17]

6 Schéma a popis testovací platformy

Uspořádání jednotlivých komponent je zřejmé z následujícího obrázku.



Obrázek 8: Schéma testovací platformy

Testování mobilní sítě je v mém případě založeno na přesměrování telefonních hovorů z telefonní sítě mobilního operátora na pobočkovou ústřednu Asterisk. Po vytočení specifického telefonního čísla, respektive rozsahu čísel, se lze dovolat právě na tuto ústřednu. Pokud je volání uskutečněno z mobilního telefonu, hovor nejprve prochází infrastrukturou mobilního operátora, který tento hovor přeměruje do ústředny umístěné na katedře telekomunikační techniky na ČVUT. Hovor je pak dále přeměrován pomocí protokolu IAX do mé domácí ústředny, kde probíhá ukládání a vyhodnocení všech testovaných parametrů. Pro testování jsem vytvořil několik typů testů, ty budou detailně popsány v jednotlivých kapitolách práce.

Pro testování bylo třeba využít kromě mobilního telefonu a pobočkových ústředěn Asterisk ještě dalších komponent. Pro automatické sestavování hovorů ze strany mobilního telefonu je využita aplikace, která slouží pro automatizaci telefonu ve všech oblastech, nejen k obsluze telefonních hovorů. Lze pak jednoduše zvolit, jak často bude telefon vytáčet určité telefonní číslo. Systém Android nepodporuje přehrávání zvukových souborů do mikrofону telefonu, tedy tak, aby byl volané straně přehrán zvukový soubor, stejně tak jako je „přehráván“ hlas, který je zachycen mikrofómem. Při volání na mobilní telefon je také velmi obtížné přehránit zvuku, který je generován volající stranou, a to z důvodů funkce potlačení echa, která je implementována ve většině současných mobilních telefonů. Z těchto důvodů jsem zvolil pro přehrávání předem vybraných zvukových souborů počítač RaspberryPi, který je s telefonem propojen audio kabelem tak, aby byl zvukový výstup RaspberryPi připojen do mikrofónního vstupu mobilního telefonu. Tuto kombinaci RaspberryPi a mobilního telefonu jsem doplnil o přenosnou baterii tak, aby byla testovací sada bez problémů přenosná a bylo tak možné provádět testování sítě v terénu.



Obrázek 9: Testovací přípravek

Je třeba zmínit důležitou skutečnost, a sice to, že tato platforma není specificky zaměřena na GSM síť, i při záměně mobilního telefonu za modernější, který bude podporovat novější technologie, budou veškeré parametry měřeny naprosto identicky. Jediným důvodem proč je v této práci využito jen GSM síť je fakt, že použitý mobilní telefon zkrátka nepodporuje novější technologie, například VoLTE. Změnou, která by tak byla eventuelně nutná, je změna testovacích zvuků, které jsou v testech, zejména pak v testu který určuje parametr MOS, přehrávány ústředně Asterisk. Změna by se týkala celkové kvality zvukové stopy, tedy například v případě využití technologie VoLTE by mohl být soubor ve značně lepší kvalitě, aby došlo k lepšímu využití přenosového pásma a byl tak otestován celkový potenciál novější technologie. Druhou možnou změnou by byla implementace novější verze algoritmu PESQ, doporučení P.863, dostupného opět na webu ITU-T.

Spojení mobilního telefonu s RaspberryPi je realizováno přes Wi-Fi síť vysílanou USB adaptérem z RaspberryPi a přehrávání zvukových souborů je pak řízeno protokolem SSH.

Veškeré zpracování a vyhodnocení testů se odehrává v počítači, kde je spuštěna má pobočková ústředna. Na tomto počítači je dále nainstalován webový server Apache, který automaticky interpretuje naměřená data na webovou stránku. Jedním z nejdůležitějších sledovaných parametrů je hodnota MOS (Mean Opinion Score), která se stanovuje právě pro jeden druh testovacího hovoru. Tato hodnota je stanovena pomocí algoritmu PESQ vydaným ITU-T.

6.1 Vliv IP sítě na naměřená data

Za zmínku pak stojí kvalita paketové sítě, přes kterou všechny hovory částečně procházejí. Její kvalita by mohla významně ovlivnit naměřené výsledky. Cílem práce však není získání přesných výsledků, ale vytvoření platformy, která bude schopna tyto výsledky získat, nicméně pro reálné nasazení je tento fakt nutné zohlednit.

Pokud budeme uvažovat část IP sítě, kterou je připojené ČVUT, domnívám se, že můžeme vyloučit jakékoliv zpomalení či ztrátovost paketů vzhledem k tomu, jak kvalitní tato síť bezpochyby je. Tato síť je pak připojena přímo do NIXu [18], kam je zároveň připojený i můj poskytovatel připojení k internetu. Tedy část sítě, která vede k mé pobočkové ústředně je se sítí ČVUT propojená asi nejlépe, jak by pro realizaci této práce mohla být. Vše shora uvedené je znázorněno ve výpisu programu Traceroute níže, který zobrazuje směrovače na cestě od mé ústředny k ústředně druhé, umístěné ve škole.

```

tracert to 147.32.***.*** (147.32.***.***), 30 hops max, 60 byte packets
 1      10.0.0.1          (10.0.0.1)    0.328 ms  0.406 ms  0.440 ms
 2      volmanova-lan.czela.czf (10.93.23.1)  .408 ms   3.623 ms  3.808 ms
 3      qbyt-ext.czela.czf    (10.93.112.1) 3.974 ms  4.146 ms  4.187 ms
 4      rum-vlan253.czela.czf  10.93.253.5   4.378 ms  5.212 ms  5.204 ms
 5      10.93.1.13          10.93.1.13    5.997 ms  5.405 ms  5.972 ms
 6      spkfree-vl3828-nfx1.nfx.cz 78.108.106.50 6.424 ms  3.609 ms  3.873 ms
 7      nix2-20ge.cesnet.cz    91.210.16.190 5.090 ms  6.375 ms  3.439 ms
 8      r92-r40.cesnet.cz     195.113.156.122 5.450 ms  5.603 ms  5.854 ms
 9      r92-cvut.cesnet.cz    195.113.144.174 4.650 ms  4.879 ms  4.895 ms
10      147.32.252.106        147.32.252.106 3.316 ms  3.995 ms  4.056 ms
11      *****.feld.cvut.cz    147.32.***.*** 4.336 ms  5.746 ms  4.563 ms

```

Tabulka 3: Traceroute mezi Asterisky

Celkové zpoždění na trase se v průměru pohybuje kolem 4 ms a tato linka je pak bezpochyby pro přenos telefonních hovorů vhodná.

6.2 Příprava Raspberry Pi

K přehrávání testovacích zvuků bylo nutné použít externí zdroj. Nejvhodnější dostupnou variantou bylo použití mini počítače Raspberry Pi, konkrétně Raspberry Pi 2 Model B a standardní distribuci pro tuto platformu Raspbian Jessie. První aplikací, kterou bylo nutno doinstalovat, byl multimediální přehrávač VLC, který je možné ovládat i z prostředí systémového terminálu, tedy i přes protokol SSH. Pro vytvoření WiFi přístupového bodu slouží USB adaptér Edimax EW-7111 USn. Ke zprovoznění přístupového bodu bylo ještě potřeba doinstalovat DHCP server, který Raspberry v základní distribuci nepodporuje, konkrétně se jedná o balíček `isc-dhcp-server`. Druhým instalovným balíčkem je `hostapd`, což je daemon, který USB adaptér nakonfiguruje jako přístupový bod. Detailní popis konfigurace je možno nalézt v [19].

6.3 Příprava mobilního telefonu

K testování je použit mobilní telefon Sony Xperia Miro s verzí systému Android 4.0.4 Ice Cream Sandwich. V telefonu je vynucené použití pouze GSM sítě. Pro obsluhu telefonních hovorů je použita aplikace Tasker [20], která je dostupná ze standardního obchodu Google Play. Jak již bylo uvedeno výše, v úvodu této kapitoly, tato aplikace slouží pro automatizaci mobilního telefonu ve všech směrech. Možností, jak tuto aplikaci použít je nepřeberné množství. Zároveň je možné doinstalovat několik takzvaných pluginů, které funkčnost aplikace ještě dále rozšiřují. V této práci je použit plugin pro SSH, který umožňuje posílat

příkazy SSH serveru, tak je realizována komunikace s Raspberry Pi. Jednotlivá nastavení této aplikace pro testovací hovory budou uvedena v příloze této práce. Ještě bych rád uvedl, že by pravděpodobně bylo efektivnější vyrobit mobilní aplikaci přímo na míru této platformě, nicméně mám s programováním pro platformu Android nulové zkušenosti. Výroba takové aplikace by pro mne byla složitější a výrazně náročnější než celá tato práce s využitím volně dostupné, již hotové aplikace Tasker.

7 Testované parametry

Podle zadání diplomové práce mají být testovanými parametry zejména doby sestavení hovorů, kvalita spojení a úspěšnost volání. Při návrhu metod, jak jednotlivé parametry změřit, jsem se musel z velké části přizpůsobit možnostem, které nabízí ústředna Asterisk a mobilní telefon. Jak již bylo uvedeno v kapitole 6, je i při pokročilých znalostech programování pro platformu Android velmi složité přimět telefon k určitým akcím souvisejících s automatickou obsluhou telefonních hovorů. Využil jsem tedy volně dostupnou aplikaci Tasker popsanou v kapitole 6, která nabízí funkci plánovače, stejně tak jako například program Crone v unixových operačních systémech. S touto aplikací se mi podařilo bez zásahu lidské ruky ovládat telefonní hovory, a to včetně jejich plánování v předem definovaných časech. Z důvodu skutečnosti, že ústředna Asterisk, respektive stroj na kterém je spuštěna, je pro vyhodnocení testů mnohem lépe použitelný, snažil jsem se hovorové testy sestavit tak, aby k jejich vyhodnocení docházelo právě na straně Asterisku.

7.1 Test dostupnosti - úspěšnosti sestavení hovoru

K ověření dostupnosti mobilního telefonu v zásadě postačuje pouze to, že po vytočení telefonního čísla dostane volající oznamovací, či jiný, tón, který vypovídá o tom, že se na volaný telefon sestavilo spojení úspěšně. Test je spouštěn ze strany ústředny, která sestavuje hovor ve směru k testovacímu mobilnímu telefonu. Ten je pomocí aplikace Tasker nastaven tak, že všechny hovory přijímá automaticky bez zásahu lidské ruky. To, spolu s automatickým sestavováním telefonních hovorů z ústředny, pro ověření dostupnosti postačuje. Pobočka pro tento hovor vypadá následovně.

```
exten => _0X.,1,Dial(IAX2/k332/${EXTEN:1})
same => n,Set(CDR(typeoftest)=Dostupnost)
```

Tato pobočka obsluhuje v zásadě všechny hovory, kdy volané číslo začíná číslicí nula, a směřuje je na IAX trunk, tedy na ústřednu umístěnou na ČVUT. Znaky `_X.` zajišťují, že není směrováno jen jedno konkrétní telefonní číslo, ale všechna čísla, která začínají nulou a jsou libovolně dlouhá. Parametr `${EXTEN:1}` pak zajistí, že telefonní číslo, které je předáno druhé ústředně umístěné na druhé straně trunku, bude o první číslici, v tomto případě o nulu, zkráceno. Jedná se o obecný způsob, jakým lze realizovat hovory mimo lokální ústřednu. Analogie je stejná jako u dřívějších pevných linek, kdy pro vytočení čísla, které patřilo účastníkovi připojenému k téže ústředně, nebylo potřeba volit předvolbu. Nejbližší ústředna pak sestavila hovor pouze na lokální úrovni a nesměrovala ho do dalších ústředn.

Druhý řádek pobočky pak do souboru CDR (viz kapitola 5) zapíše do pole `typeoftest` textový řetězec „Dostupnost“ tak, aby bylo možné identifikovat druh testu. Záznamy o všech hovorových testech se ukládají do stejného souboru. Ten je dále předán skriptu, který pro každý druh testu vytvoří nový soubor s odpovídajícími údaji. Výstup každého úspěšného hovoru je pak následující.

```
Volany:797990104, Start:2016-05-11,13:45:38 Prijato:2016-05-11 13:45:44
Celkem[s]:1.422263, Info:ANSWERED
```

Neúspěšný hovor pak generuje následující výstup.

```
Volany:0797990104 Start:2016-05-15 14:42:49 Prijato: Ukonceno: 2016-05-15
14:42:49
Celkem[s]:0.000000 Info:FAILED
```

Aby Asterisk zaznamenával údaje i pro neúspěšné hovory, je potřeba nastavit v konfiguračním souboru `/etc/asterisk/cdr.conf` volbu `unanswered` na `yes`. Vyhodnocení probíhá až po stisknutí tlačítka na webové stránce, kam se data interpretují. Výstup vyhodnocení je poměrně strohý, nicméně pro kontrolu naměřených dat postačující. Například takový:

```
Celkovy pocet volani: 4 Celkovy pocet uspesnych volani 3
Procentualni uspesnost: 75 %
```

U tohoto testu není bohužel možné určit přesnou příčinu případné nedostupnosti mobilního telefonu, pokud nastane chyba až na rádiovém rozhraní. V ostatních případech lze zjistit příčinu neúspěšnosti pomocí položky `HANGUPCAUSE` v CDR souboru. Popis číselných symbolů, které tato proměnná vrácí je popsán v příloze práce. Jednotlivé symboly pak odpovídají různým chybovým hláškám.

Položka `HANGUPCAUSE` tedy dokáže zobrazit více informací o případné chybě než položka `status`, která je umístěna v souboru CDR standardně již po instalaci ústředny.

7.2 Test DTMF

Při určení kvality spojení jsem zpočátku využil jedné z funkcí, které Asterisk nabízí. Protože lze ústřednu využít mimo jiné i jako hlasový automat, který je možné ovládat tónovou volbou, tedy volbou DTMF [viz [23]], podporuje Asterisk i detekci těchto tónů.

Pokud je tedy Asterisku během hovoru přehrán některý tón DTMF, lze ho zaznamenat pomocí aplikace `Read`. Pro účely testování jsem, pomocí skriptu pro vývojové prostředí Matlab uvedeného v příloze práce, vygeneroval posloupnost tónů jednotlivých číslic, kterou jsem uložil jako zvukový soubor a ústředně ho přehrával. Pomocí vhodně definované pobočky pak lze vyhodnotit detekované číslice.

```
exten => 5,1,Answer
same => n,Read(dtmfinput,,25,skip,1,19)
same => n,Set(CDR(typeoftest)="DTMFtest")
same => n,Set(CDR(dtmfinput)=${dtmfinput})
same => n,GotoIf("${dtmfinput}"="0123456789876543210"?OK:chyba)
same => n(OK),Set(CDR(spravnost)=OK)
same => n,Goto(end)
same => n(chyba),Set(CDR(spravnost)=chyba)
same => n(end),Hangup
```

Aplikace `Read` má několik parametrů, kterými lze ovlivnit její chování. Prvním takovým parametrem je proměnná, do které bude vstup ukládán. Dalším (vynechaným) parametrem je možnost určení zvukového souboru, který by byl před detekcí čísel volajícímu přehrán. Následuje počet číslic, které má Asterisk očekávat. Záměrně je volen větší počet číslic, než jaký je ve skutečnosti ústředně přehráván, a to z důvodu možného chvilkového výpadku při přenosu tónové volby. Asterisk při takovém výpadku může jednotlivou číslici detekovat vícekrát. Pokud dojde k úplnému výpadku spojení, volba `skip` aplikaci `Read` pak ukončí. Posledními parametry jsou počet pokusů na detekci volby a časové omezení doby, po kterou má ústředna tóny detekovat. Po této aplikaci následuje zapsání názvu testu do položky `CDR typeoftest` (viz kapitola 8) a zapsání volených číslic do položky `dtmfinput`. Tyto dvě položky jsou následně vypisovány do přehledu o provedeném testu. Na následujícím řádku je pak definována podmínka pro určení správnosti volené sekvence. Pokud se obsah proměnné, do které aplikace `Read` ukládá volená čísla, shoduje s uvedeným řetězcem, přejde se v rámci pobočky na řádek s prioritou `OK`, pokud se volba neshoduje, přejde Asterisk na prioritu `chyba`. Po tomto vyhodnocení se do `CDR` položky `spravnost` zapíše, zda detekce proběhla správně, či ne.

Celý test je iniciován z mobilního telefonu, který po vytočení testovacího telefonního čísla vyšle přes Wi-Fi síť SSH příkaz pro přehrávání zvukového souboru do RaspberryPi, s kterým je propojen pomocí audio kabelu, jak je uvedeno v kapitole 6. Výsledkem tohoto testu je pro jednotlivý hovor jeden řádek v logu `CDR`. Výstup tak může vypadat

následovně.

```
Volajici:797990104
Prijato:2016-05-12 03:20:09
Ukonceno:2016-05-12 03:20:53
Celkem[s]:44.49 Info:ANSWERED
Volena sekvence:123456789 Spravnost:OK
```

Tento test však nemá nijak velkou vypovídající hodnotu o kvalitě spojení. Lze ho využít pro detekci krátkodobých výpadků, protože pak se ve volené sekvenci objeví některé z čísel vícekrát, nebo vůbec.

7.3 Test doby sestavení hovoru

Určit dobu sestavení hovoru je bez přístupu k prvkům GSM sítě poněkud obtížné. Tato doba je definována (v případě, kdy je doba měřena z mobilního telefonu do pevné sítě) jako průměr naměřených dob mezi okamžikem, kdy mobilní stanice požádá o přidělení komunikačního kanálu, a okamžikem, kdy je mobilní stanicí přijata zpráva alerting, která signalizuje, že byl požadavek na spojení odeslán protější straně hovoru. V této práci je však doba měřena jako rozdíl okamžiku, kdy mobilní telefon požádá BTS o přidělení komunikačního kanálu, a okamžiku, kdy Asterisk zaregistruje příchozí hovor. Vzniká zde tedy malá odchylka a ve výsledku by naměřené hodnoty měly být o několik okamžiků kratší, nicméně s dostupnými prostředky není možné dosáhnout přesnějších výsledků, stejně tak není možné zcela přesně určit časové okamžiky jednotlivých signalizačních zpráv, které při sestavení hovorů probíhají. Pokud se však na situaci podíváme z pohledu účastníka, jeho vnímání doby sestavení hovoru je přibližně stejné, jaké je měřeno v tomto testu.

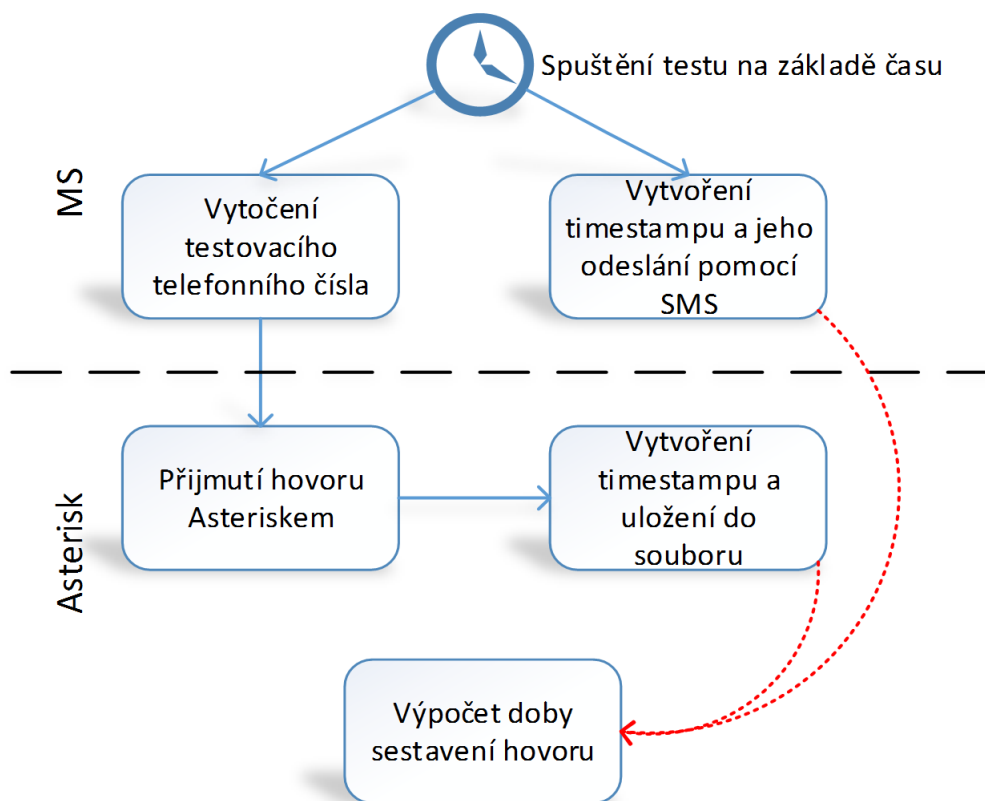
Při měření doby sestavení hovoru z mobilního telefonu do pevné sítě, tedy do ústředny Asterisk, bylo nutné přenést do této ústředny přesný časový údaj o začátku hovoru. K tomu bylo využito SMS zpráv a konfigurace v mobilní aplikaci Tasker, která umožňuje při začátku testovacího hovoru automatické odeslání SMS zprávy, ve které je zaznamenán přesný čas v unixovém formátu, tedy v počtu sekund uplynulých od 1.1.1970. Pro tento účel tak musí telefon být přesně časově synchronizovaný s časem ústředny Asterisk. V systému Ubuntu lze časové synchronizace dosáhnout pomocí balíčku `ntp` a následné modifikaci souboru `/etc/ntp.conf`. V tomto souboru jsem přidal na odpovídající řádek nový NTP server `ntp.nic.cz`.

Po zadání příkazu `timedatectl` je pak vidět synchronizace pomocí NTP serveru.

```
Local time: Ne 2016-05-15 22:41:34 CEST
Universal time: Ne 2016-05-15 20:41:34 UTC
RTC time: Ne 2016-05-15 20:41:34
Timezone: Europe/Prague (CEST, +0200)
NTP enabled: yes
NTP synchronized: yes
```

Na straně mobilního telefonu je synchronizace zajištěna pomocí mobilní sítě operátora. Po kontrole pomocí volně dostupné aplikace jsem zjistil, že telefon je oproti přesnému času o zhruba 150 ms napřed. Tato hodnota je tedy odečtena při výpočtu konečné doby sestavení hovoru. Protože Asterisk není schopen přijímat SMS zprávy z pevné sítě, do které je připojený (přesněji řečeno chybí tato podpora v síti mobilního operátora, Asterisk je z jeho pohledu pevný telefon, na který operátor SMS zprávy nedoručuje), byl pro příjem SMS zpráv použit USB modem Huawei E3131 se SIM kartou, která se přihlašuje do mobilní sítě. Modem však není přímo softwarově propojen s ústřednou, pro příjem SMS zpráv je použita aplikace `gammu`, která je jednou z nejrozšířenějších aplikací pro správu mobilních telefonů a modemů přes USB rozhraní, a v prostředí terminálu poskytne uživateli mnoho funkcí. Přijaté zprávy se standardně ukládají do složky `/var/spool/gammu/inbox/`. Příjem zpráv je řízen `daemonem`, příchozí zpráva se tak v systému objeví okamžitě po jejím doručení.

Scénář testu je uveden na následujícím obrázku.



Obrázek 10: Scénář testu doby sestavení hovoru

Obě časové známky jsou generované v unixovém formátu, včetně milisekund. K výpočtu doby sestavení hovoru tak stačí pouze tyto dva časy odečíst (včetně chyby) a posunout desetinnou čárku tak, aby výsledné číslo odpovídalo údajům v sekundách. Tato funkce je vykonávána skriptem, který se spouští každou minutu a vyhodnocuje všechny druhy testů společně. Část tohoto skriptu, která zajišťuje výpočet, je uvedena v následující kapitole. Celý skript je uveden v příloze práce.

Pobočka pro tento test vypadá následovně.

```
exten => 6,1,Answer
same => n,System(date +%s%3N >> /HomeDir/scripts/answered.txt)
same => n,Set(CDR(typeoftest)="setuptime")
same => n,Wait(3) same => n,Hangup()
```

Za zmínku pak stojí rovněž popis druhého řádku pobočky. Asterisk umožňuje spouštět systémové příkazy pomocí aplikace System. Argument v závorce je tak příkaz pro prostředí linuxového terminálu, který vygeneruje časovou známku a uloží ji do dočasného souboru `answered.txt`.

Výstup testu vypadá následovně.

```
Volajici:797990104 Prijato:2016-05-15 23:20:04
Info:ANSWERED Setup time[s]= 4.335
```

7.4 Test SMS zpráv

Pro realizaci předchozího testu je nutné posílat z mobilního telefonu SMS zprávy, které vždy obsahují časovou známku okamžiku, kdy jsou odeslány, respektive okamžiku, kdy začal hovor. Vzhledem k tomu, že aplikace gammu pro zpracování SMS zpráv nabízí možnost spustit uživatelem vytvořený skript ve chvíli, kdy je přijata nová textová zpráva, považoval jsem za vhodné realizovat i test, který měří dobu příjmu SMS zpráv. Skript, který se spouští po příjmu SMS zprávy je následující.

```
#!/bin/bash
date +%s%3N >> /HomeDir/scripts/received.txt
```

Vytváří tedy další časovou známku, podle které je ve skriptu pro zpracování všech výsledků vypočtena doba příjmu SMS. Část tohoto skriptu je následující. Z důvodu provázanosti s předchozím testem, tedy testem doby sestavení hovoru, je uvedena i část pro předchozí test.

```
{
if [ ! -f /var/spool/gammu/inbox/*0797990104* ]; then
echo "\SMS jeste nedorazila!"
exit 0
fi
}

originated=$(cat /var/spool/gammu/inbox/*0797990104*)
answered=$(cat /HomeDir/scripts/answered.txt)
recsms=$(cat /HomeDir/scripts/received.txt)
```

```

s=$(( $answered - $originated - 150 ))
r=$(( $recsms - $originated - 150 ))
setuptime=$(printf %d.%d ${s%????} ${s#s%????})
smsdelivery=$(printf %d.%d ${r%????} ${r#r%????})
log=$(cat /HomeDir/vysledky/setuptime.csv)
paste <(echo "$log") <(echo "$setuptime")
        >> /HomeDir/vysledky/SETUPtime.csv
echo "$smsdelivery" >> /HomeDir/vysledky/SMStime.txt
rm /HomeDir/vysledky/setuptime.csv
rm /HomeDir/scripts/answered.txt
rm /var/spool/gammu/inbox/*
rm /HomeDir/scripts/received.txt

```

Vzhledem k tomu, že skript, který zpracovává výsledky je spuštěn každou minutu, je před uvedenou částí skriptu ještě dána podmínka, která kontroluje, zda už došlo k příjmu SMS zprávy z testovacího telefonního čísla. Pokud k příjmu zprávy došlo, dojde k vykonání uvedeného skriptu, pokud ne, je počkáno až na další spuštění vyhodnocovacího skriptu. V prvním řádku po podmínce proběhne načtení časové známky, která je obsažena v příchozí textové zprávě, v dalším řádku je načtena časová známka, která odpovídá okamžiku příjmu hovoru určeného pro test doby sestavení, a v dalším řádku je pak načtena časová známka okamžiku, kdy byla přijata textová zpráva. Poté následují výpočty dob sestavení hovoru a příjmu textové zprávy, včetně odečtení 150 milisekund, důvod odečtení je uveden výše v této kapitole. Další tři řádky slouží pro načtení souboru CDR a připojení vypočtené doby do poslední buňky tabulky a vytvoření (při prvním testu) nebo přidání (při každém dalším testu) časového údaje do souboru s časy dob příjmu SMS zpráv. Poté jsou smazány všechny pomocné soubory a přijatá SMS zpráva.

7.5 Test kvality spojení

Jak již bylo uvedeno v kapitole 4, je v této práci použit algoritmus PESQ vydaný Mezinárodní telekomunikační unií. Tento algoritmus je možné získat přímo na internetových stránkách ITU. [9]

Po rozbalení staženého archivu se soubory, je potřeba zkopírovat soubory ze složky source do libovolné složky v souborovém systému. Kompilaci aplikace je pak možné provést následujícím příkazem.

```
gcc -o PESQ *.c -lm
```

Po zkompileování se ve složce vytvoří soubor PESQ, který je možné spustit příkazem `./PESQ`. Jako vstupní parametr je nutné uvést dva zvukové soubory, degradovaný a originální, a vzorkovací frekvenci, s jakou jsou oba soubory nahrány. Algoritmus pak vrací následující výstup.

REFERENCE	DEGRADED	PESQMOS	SAMPLE_FREQ	CRUDE_DELAY
pesq.wav	pesq-rec.wav	3.774	8000	6.2040

Tabulka 4: Výstup algoritmu PESQ

Pro získání hodnoty PESQ tak stačí pouze jedna buňka z tabulky. Poté, co je pro každý testovací hovor vypočtena hodnota PESQ, která je vypočtena těsně před koncem hovoru, jak je vidět v níže uvedené pobočce pro tento test, je vyčkáno na spuštění vyhodnocovacího skriptu, který, dle rovnice uvedené v kapitole 4, dopočítá konečnou hodnotu parametru MOS. Výsledný záznam, který je zobrazen na webové stránce, je tak spojením logu generovaného Asteriskem a proměnné, která je vypočtena během vyhodnocení.

Pobočka k tomuto testu je následující.

```

exten =>4,1,Answer()
same => n,Set(CDR(typeoftest)=MOSStest)
same => n,Monitor(wav,/HomeDir/pesq/nahravky_ast/
                ${STRFTIME(,,record%e%m%H:%M)},o)
same => n,Wait(15)
same => n,StopMonitor()
same => n,System(cp /HomeDir/pesq/nahravky_ast/rec*
                /HomeDir/pesq/MOSrecord)
same => n,System(mv /HomeDir/pesq/nahravky_ast/rec*
                /HomeDir/pesq/temp)
same => n,System(cd /HomeDir/pesq/temp/ &&
                ./PESQ +8000 pesq.wav rec*)
same => n,System(rm /HomeDir/pesq/temp/rec*)
same => n,Hangup()

```

V tomto testu je využita aplikace Monitor, která umožňuje nahrávat hovor pro danou pobočku. Parametr „o“ pak určuje, že bude nahrávána pouze příchozí zvuková stopa. Standardně generuje aplikace Monitor pro každý hovor dva zvukové soubory, rozlišené pomocí řetězce „in“ a „out“ v názvu souboru. Název souboru pak odpovídá datu a

času v okamžiu příchodu hovoru. Výsledný soubor pak může vypadat takto: `record 70515:35-in.wav`. Dále jsou opět použity aplikace `System`, které zpracovávají nahraný soubor a počítají výslednou hodnotu MOS.

8 Interpretace výsledků

Dle zadání diplomové práce mají být naměřené výsledky interpretovány na WWW stránce. K tomuto účelu jsem využil známý webový server Apache. Jeho konfigurace je pro potřeby této práce snadná. Výsledky jsou převáděny ze souboru CDR (viz kapitola 5) na html kód, který je zobrazen pomocí PHP skriptu. Pro každý druh testu je na stránce vytvořena vlastní záložka. Dále umí webová stránka zálohovat naměřená data, tato funkce zároveň slouží pro vymazání aktuálně zobrazovaných naměřených dat tak, aby bylo snadno možné zahájit nové měření a nebylo tak potřeba manuální přihlašování na server a přesouvání souborů s naměřenými daty. Přiznávám, že tato stránka je první, jakou jsem kdy vytvořil, proto může být z pohledu zkušeného programátora řešena poměrně složitě a ne zcela efektivně, nicméně se domnívám, že celkově je WWW stránka přehledná, použitelná a hlavně funkční a pro účely zadání diplomové práce plně dostačující. Vzhledem k tomu, že naměřená data jsou vždy ukládána v textových souborech, které mají stejnou strukturu, nepoužil jsem při jejich zpracování žádný databázový systém. Proto například chybí funkce řazení naměřených dat podle určitých hodnot, například řazení od nejkratší doby sestavení hovoru po nejdelší, atd. Při tvorbě webové stránky bylo čerpáno z [21, 22].

8.1 Konfigurace `cdr_custom`

Jak bylo uvedeno v kapitole 5, Asterisk pro každý hovor generuje záznam do souboru označovaného CDR. Ten existuje jak ve standardní podobě, kterou má Asterisk již po instalaci předdefinovanou, tak i v podobě „`custom`“, kterou si může administrátor přizpůsobit dle svých požadavků.

Má konfigurace vlastního CDR záznamu vypadá následovně.

```
def_hovory.csv => typ testu:${CDR(typeoftest)},
Volajici:${CDR(src)},
Start:${CDR(start)},
Prijato:${CDR(answer)},
Ukonceno:${CDR(end)},
Celkem[s]:${CDR(billsec,f)},
```

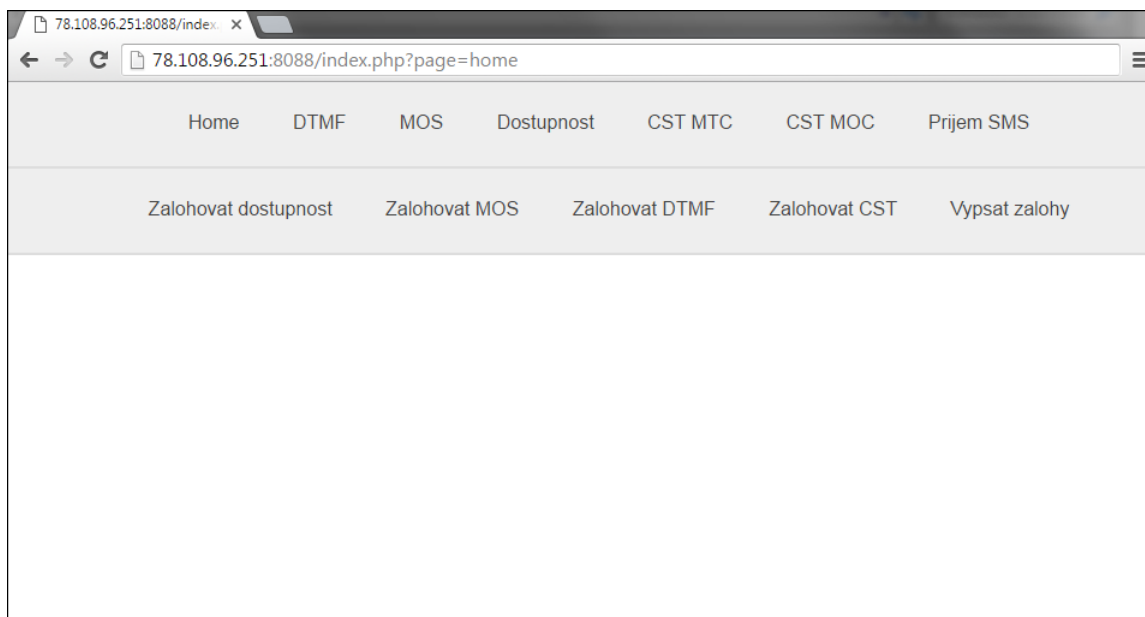
```
Info:${CDR(disposition,r)},
Volena sekvence:${CDR(dtmfinput)},
Spravnost:${CDR(spravnost)},
MOS=,
Setup time [s]=,
cause:${CDR(userfield)}
```

Generovaný soubor má tedy název `def_hovory.csv`. Mimo položky, které jsou uvedené v kapitole 5, se zde vyskytují položky, které byly popsány u jednotlivých testů. Jednotlivé údaje jsou odděleny čárkou, ta je velmi důležitá pro následné zpracování, slouží totiž jako oddělovač jednotlivých „buněk“. Proto například není u položek „MOS“ a „Setup time“ žádný záznam. Tento text se pak vypíše před údaj, který je doplněn na desátou, respektive jedenáctou pozici a je tak možné tento doplněný údaj rozpoznat. Doplnění je realizováno shellovým skriptem.

Spolu s tímto CDR souborem je generován ještě soubor druhý. Ten je přizpůsoben zobrazení na webové stránce a názvy položek jsou vloženy do html značek, pro které je zakázáno zobrazení. Při zobrazení v internetovém prohlížeči se tak nezobrazují názvy jednotlivých položek u každé jednotlivé buňky, ale jsou zobrazeny v hlavičce výsledné tabulky.

8.2 Struktura webové stránky

Webová stránka je koncipována poměrně jednoduše. V horní části je menu, pomocí kterého lze přepínat mezi výsledky jednotlivých testů. Po kliknutí se zobrazí tabulka s výsledky daného testu. Při začátku nového testování je možné zálohovat doposud zobrazované výsledky pomocí tlačítek v podmenu na hlavní stránce. Tato operace je realizována spuštěním shellového skriptu, který přesune soubory v souborovém systému.



Obrázek 11: Úvodní webová stránka

Volajici	Prijato	Ukonceno	Celkem [s]	Info	MOS
797990104	2016-05-21 10:40:05	2016-05-21 10:40:32	27	ANSWERED	3.455
797990104	2016-05-21 10:50:05	2016-05-21 10:50:32	27	ANSWERED	3.769
797990104	2016-05-21 11:10:05	2016-05-21 11:10:32	27	ANSWERED	0.992
797990104	2016-05-21 11:20:04	2016-05-21 11:20:31	27	ANSWERED	3.135
797990104	2016-05-21 11:30:06	2016-05-21 11:30:33	27	ANSWERED	3.345
797990104	2016-05-21 11:40:05	2016-05-21 11:40:32	27	ANSWERED	3.757
797990104	2016-05-21 11:50:07	2016-05-21 11:50:34	27	ANSWERED	3.391

Obrázek 12: Výsledky testu kvality spojení

8.2.1 Převod CDR souborů

Převod CDR souboru, který odpovídá testu kvality spojení, je proveden následujícím skriptem.

```
<?php
echo "<table cellpadding='5' border='1'>\n\n";
echo "<tr><th>Volajici</th><th>Prijato</th><th>Ukonceno</th>
      <th>Celkem [s]</th>";
$f = fopen("HomeDir/vysledky/finalmos.csv", "r");
while (($line = fgetcsv($f)) !== false) {
    echo "<tr>";
    foreach ($line as $cell) {
        echo "<td>" . $cell . "</td>";
    }
    echo "</tr>\n";
}
fclose($f);
echo "\n</table>"
?>
```

Skript generuje html tabulku, do které přepisuje údaje ze CDR souboru. Zároveň pro každý test vytvoří hlavičku tabulky, kterou je nutné definovat ručně. Současně je možné zviditelnit okraje tabulky nebo přizpůsobit rozestupy mezi jednotlivými buňkami.

Vzhledem k tomu, že skripty pro ostatní testy jsou v podstatě totožné, nebudu je uvádět všechny, ale budou uvedeny na příloženém CD k této práci.

9 Závěr

Cílem této diplomové práce bylo vytvořit platformu pro testování mobilních sítí. Platforma je sestavena pro linuxové operační systémy, jádrem celé platformy je pak pobočková ústředna Asterisk. Při realizaci samotných testů jsem se musel přizpůsobit možnostem, které pobočková ústředna Asterisk nabízí. Další omezení pak představují možnosti mobilního telefonu platformy Android. S ohledem na tyto uvedené důvody bylo při realizaci potřeba užití dalších hardwarových prvků, konkrétně zdroje přehrávání testovacích zvuků či USB modemu pro příjem textových zpráv na serveru, na kterém je spuštěn Asterisk. Zdroj zvuku by nemusel být součástí této platformy, pokud by se platforma dostala do rukou zkušenému programátorovi mobilních aplikací pro systém Android. Věřím, že po určitých zásazích do tohoto systému by bylo možné přehrávat testovací zvuky přímo z telefonu, což by bylo zajisté více efektivní řešení, nicméně fakt, že v této práci není využito vlastní naprogramované aplikace, nikterak neubírá na celkové funkčnosti a použitelnosti této testovací platformy.

Samotné testování je prováděno pomocí pěti testů, kterými lze stanovit několik klíčových výkonostních parametrů GSM sítě. Konkrétními parametry jsou doba sestavení hovoru, dostupnost mobilní stanice, kvalita sestaveného spojení nebo doba příjmu SMS zpráv. Tyto parametry jsou pak reprezentovány na webové stránce, která je rozdělena do několika částí podle testovaných parametrů.

Testovací platforma by mohla být v určitých směrech vylepšena. Jednak, jak již bylo zmíněno, naprogramováním vlastní mobilní aplikace nebo také použitím některého dostupného databázového systému, který by dokázal lépe pracovat s naměřenými hodnotami a mohl je tak například řadit podle naměřených parametrů nebo by jednotlivé testy, respektive místa, kde byly testy provedeny, zároveň dokázal vynášet na mapový podklad tak, aby byla vidět místa, která mají horší pokrytí mobilním signálem. Podle zadání práce nemá být její součástí vyhodnocení celkové kvality mobilní sítě, nicméně jsem ze zvědavosti provedl zhruba třicet různých testů z jedoucího automobilu, abych tak mohl pozorovat rozdíly v naměřených hodnotách oproti situaci, kdy je testovací přípravek umístěn na stole, nehýbe se a úroveň přijmaného signálu sahá k nejlepším možným hodnotám. Rozdílnost ve výsledcích mezi zmíněnými příklady je viditelná na první pohled. Dalším vylepšením by dále mohlo být rozšíření testovaných parametrů do oblasti přenosu dat.

Osobně pro mě byla práce velkým přínosem, za největší přínos pokládám seznámení se s pobočkovou ústřednou Asterisk a dále pak celkové zlepšení se v práci s linuxovým operačním systémem, zejména v prostředí terminálu, protože veškeré práce na serveru s ústřednou probíhaly přes vzdálený přístup pomocí SSH protokolu.

Reference

- [1] ČESKÝ STATISTICKÝ ÚŘAD. Informační společnost v číslech - 2015 [online]. 2015, , 1 [cit. 2016-05-22]. Dostupné z: download.mpo.cz/get/52735/60041/632131/priloha001.pdf
- [2] MISHRA, Ajay R. Fundamentals of cellular network planning and optimisation: 2G, 2.5G, 3G– evolution to 4G. Hoboken, NJ: Wiley, c2004. ISBN 04-708-6267-X.
- [3] EBERSPÄCHER, J. GSM: architecture, protocols and services. 3rd ed., English lang. ed. Chichester, U.K.: Wiley, c2009. ISBN 04-700-3070-4.
- [4] SUDHINDRA, K.R. a V. SRIDHAR. Root cause detection of Call drops in live GSM network. TENCON 2011 - 2011 IEEE Region 10 Conference [online]. IEEE, 2011, , 440-444 [cit. 2016-05-23]. DOI: 10.1109/TENCON.2011.6129142. ISBN 978-1-4577-0255-6. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6129142>
- [5] KYRIAZAKOS, S., V. GKROUSTIOTIS, C. KARAMBALIS, C. KECHAGIAS, N. PAPAOUKAKIS, D. NIKITOPoulos a G. KARETSOS. Performance evaluation of GSM and GPRS systems based on measurement campaigns: proceedings : WPMC 02, October 27-30, 2002, Honolulu, Hawaii. The 5th International Symposium on Wireless Personal Multimedia Communications [online]. IEEE, 2002, , 1167-1171 [cit. 2016-05-23]. DOI: 10.1109/WPMC.2002.1088362. ISBN 0-7803-7442-8. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1088362>
- [6] OYETUNJI, Oyedemi. Improving Call Setup Success Rate in GSM service area using RF optimisation. 2014 11th International Conference on Electronics, Computer and Computation (ICECCO) [online]. IEEE, 2014, , 1-4 [cit. 2016-05-23]. DOI: 10.1109/ICECCO.2014.6997554. ISBN 978-1-4799-4106-3. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6997554>
- [7] ITU-T Recommendation P.800.2: Mean opinion score interpretation and reporting [online]. Geneva, 2013 [cit. 2016-05-23]. Dostupné z <http://www.itu.int/rec/T-REC-P.800.2/en>
- [8] ITU-T Recommendation P.800.1: Mean Opinion Score (MOS) terminology [online]. Geneva, 2016 [cit. 2016-05-23]. Dostupné z <http://www.itu.int/rec/T-REC-P.800.1/en>

- [9] ITU-T Recommendation P.862: Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs [online]. Geneva, 2001 [cit. 2016-05-23]. Dostupné z <http://www.itu.int/rec/T-REC-P.862-200102-I/en>
- [10] ITU-T Recommendation P.862.1: Mapping function for transforming P.862 raw result scores to MOS-LQO [online]. Geneva, 2003 [cit. 2016-05-23]. Dostupné z <http://www.itu.int/rec/T-REC-P.862.1-200311-I/en>
- [11] Get Started [online]. 2016 [cit. 2016-05-23]. Dostupné z: <http://www.asterisk.org/get-started>
- [12] Asterisk Configuration. Voip-info.org [online]. San Diego: Voip-Info, [cit. 2016-05-23]. Dostupné z: <http://www.voip-info.org/wiki/view/Asterisk+config+files>
- [13] Asterisk config iax.conf. Voip-info.org[online]. San Diego: Voip-Info, 2012 [cit. 2016-05-23]. Dostupné z: <http://www.voip-info.org/wiki/view/Asterisk+config+iax.conf>
- [14] Asterisk config sip.conf. Voip-info.org[online]. San Diego: Voip-Info, 2016 [cit. 2016-05-23]. Dostupné z: <http://www.voip-info.org/wiki/view/Asterisk+config+sip.conf>
- [15] Asterisk config extensions.conf. Voip-info.org[online]. San Diego: Voip-Info, 2015 [cit. 2016-05-23]. Dostupné z: <http://www.voip-info.org/wiki/view/Asterisk+config+extensions.conf>
- [16] Asterisk config rtp.conf rtp.conf. Voip-info.org [online]. San Diego: Voip-Info, 2009 [cit. 2016-05-23]. Dostupné z: <http://www.voip-info.org/wiki/view/Asterisk+config+rtp.conf>
- [17] Asterisk auto-dial out. Voip-info.org [online]. San Diego: Voip-Info, 2015 [cit. 2016-05-23]. Dostupné z: <http://www.voip-info.org/wiki/view/Asterisk+auto-dial+out>
- [18] Neutral Internet eXchange. NIX.CZ – Neutral Internet eXchange [online]. Praha: NIX.CZ, 2016 [cit. 2016-05-23]. Dostupné z: <https://www.nix.cz/cs/about>
- [19] Setting up a Raspberry Pi as a WiFi access point. Adafruit learning systems [online]. Adafruit Industries, 2015 [cit. 2016-05-23]. Dostupné z: <https://cdn-learn.adafruit.com/downloads/pdf/setting-up-a-raspberry-pi-as-a-wifi-access-point.pdf>
- [20] Tasker. Google Play [online]. Google, 2016 [cit. 2016-05-23]. Dostupné z: <https://play.google.com/store/apps/details?id=net.dinglich.android.taskerm&hl=cs>

- [21] KOFLER, Michael a Bernd ÖGGL. PHP 5 a MySQL 5: průvodce webového programátora. Brno: Computer Press, 2007. ISBN 978-80-251-1813-9.
- [22] W3schools.com: THE WORLD'S LARGEST WEB DEVELOPER SITE [online]. w3schools, 2016 [cit. 2016-05-23]. Dostupné z: <http://www.w3schools.com/>
- [23] ITU-T Recommendation Q.23: Technical features of push-button telephone sets [online]. Geneva, 1988 [cit. 2016-05-23]. Dostupné z <https://www.itu.int/rec/T-REC-Q.23/en>
- [24] Hangup Cause Mappings. Asterisk wiki [online]. 2013 [cit. 2016-05-23]. Dostupné z: <https://wiki.asterisk.org/wiki/display/AST/Hangup+Cause+Mappings>
- [25] LTE. 3gpp [online]. [cit. 2016-05-23]. Dostupné z: <http://www.3gpp.org/technologies/keywords-acronyms/98-lte>
- [26] ITU-T Recommendation P.862.3: Application guide for objective quality measurement based on Recommendations P.862, P.862.1 and P.862.2 [online]. Geneva, 2007 [cit. 2016-05-23]. Dostupné z: <https://www.itu.int/rec/T-REC-P.862.3/en>

10 Přílohy

10.1 Konfigurační soubory Asterisku

10.1.1 cdr_custom.conf

```
[mappings]
;následující mapování je určené pro případné prohlížení výsledků v
;systémovém terminálu
def_hovory.csv => typ testu:${CDR(typeoftest)},
Volajici:${CDR(src)},
Start:${CDR(start)},
Prijato:${CDR(answer)},
Ukonceno:${CDR(end)},
Celkem[s]:${CDR(billsec,f)},
Info:${CDR(disposition,r)},
Volena sekvence:${CDR(dtmfinput)},
Spravnost:${CDR(spravnost)},
MOS=,
Setup time [s]=,
cause:${CDR(userfield)}

;následující mapování je upravené pro zobrazení na webové stránce
hovory.csv =><i>typ testu:</i>${CDR(typeoftest)} ,
<i>Volajici:</i>${CDR(src)},
<i>Start:</i>${CDR(start)} ,
<i>Prijato:</i>${CDR(answer)} ,
<i>Ukonceno:</i>${CDR(end)},
<i>Celkem[s]: </i>${CDR(billsec,r)},
<i>Info:</i>${CDR(disposition)},
<i>Volena sekvence:</i>${CDR(dtmfinput)},
<i>Spravnost:</i>${CDR(spravnost)},
<i>MOS=</i>,
<i>cause:</i>${CDR(HANGUPCAUSE)},
${CDR(cstmtc)}
```

10.1.2 extensions.conf

```
[general]
static=yes
writeprotect=no
priorityjumping=no
autofallthrough=yes
clearglobalvars=no
[globals]
[public]

;směrování hovorů na IAX trunk
exten => _OX.,1,Dial(IAX2/k332/${EXTEN:1},,g)

;pobočka h, která je spuštěna po spuštění aplikace hangup,
;tedy po ukončení hovoru
exten => h,1,Set(CDR(HANGUPCAUSE)=${HANGUPCAUSE})

;MOS test
exten =>4,1,Answer()
same => n,Set(CDR(typeoftest)=MOSStest)
same => n,Monitor(wav,/HomeDir/pesq/nahravky_ast/
${STRFTIME(,,record%e%m%H:%M)},o)
same => n,Wait(25)
same => n,StopMonitor()
same => n,System(cp /HomeDir/pesq/nahravky_ast/rec*
/HomeDir/pesq/MOSrecord)
same => n,System(mv /HomeDir/pesq/nahravky_ast/rec*
/HomeDir/pesq/temp)
same => n,System(cd /HomeDir/pesq/temp/ &&
./PESQ +8000 pesq.wav rec*)
same => n,System(rm /HomeDir/pesq/temp/rec*)
same => n,Hangup()

;DTMF test
exten => 5,1,Answer
same => n,Read(dtmfinput,,20,skip,1,15)
```

```

same => n, Set(CDR(typeoftest)="DTMFtest")
same => n, Set(CDR(dtmfinput)={dtmfinput})
same => n, GotoIf("${dtmfinput}"="123456789"?OK:chyba)
same => n(OK), Set(CDR(spravnost)=OK)
same => n, Goto(end)
same => n(chyba), Set(CDR(spravnost)=chyba)
same => n(end), Hangup()

;setuptime
exten => 6,1,Answer
same => n, System(date +%s%3N >> /HomeDir/scripts/answered.txt)
same => n, Set(CDR(typeoftest)="setuptime")
same => n, Wait(3) same => n, Hangup()

;test dostupnosti - nový kontext, proto je opět uvedena
;pobočka h s prioritou 1, v jednom kontextu smí taková
;pobočka být pouze jednou

[dostupnost]
exten => 797990104,1, Set(CDR(typeoftest)=Dostupnost)
same => n, Wait(3)
same => n, Set(CDR(setupmtc)={DIALEDTIME})
same => n, Hangup()
exten => h,1, Set(CDR(HANGUPCAUSE)={HANGUPCAUSE})

```

10.1.3 iax.conf

```

[general]
[k332]
type=friend
username=*****
secret=*****
auth=md5
host=147.32.***.***
context=public
qualify=yes
qualifysmoothing = yes
jitterbuffer=no
trunk=no
connectedline=yes
disallow=all

```

```
allow=g722
allow=alaw
```

10.1.4 sip.conf

```
[general]
;definice IP adres a portu pro SIP
allowguest=no
bindport=50605
externip=78.108.96.251
localnet=10.0.0.0/255.255.255.0

;ukázka založení SIP klienta
[724763523]
type=friend
qualify=yes
disallow=all
allow=g722
allow=alaw
allow=gsm
secret=tredcvbg
host=dynamic
context=public
```

10.2 Mapování Hangup Cause

1. Unallocated (unassigned) number
2. No route to specified transmit network
3. No route to destination
5. Misdialled trunk prefix (national use)
6. Channel unacceptable
7. Call awarded and being delivered in an established channel ISUP -
8. Preemption
14. QoR: ported number
16. Normal Clearing
17. User busy
18. No user responding
19. No answer from user (user alerted)
20. Subscriber absent
21. Call Rejected
22. Number changed
23. Redirected to new destination

26. Non-selected user clearing
27. Destination out of order
28. Invalid number format
29. Facility rejected
30. Response to STATUS ENQUIRY
31. Normal, unspecified
34. No circuit/channel available
38. Network out of order
41. Temporary failure
42. Switching equipment congestion
43. Access information discarded
44. Requested circuit/channel not available
50. Requested facility not subscribed
52. Outgoing call barred
54. Incoming call barred
57. Bearer capability not authorized
58. Bearer capability not presently available
65. Bearer capability not implemented
66. Channel type not implemented
69. Requested facility not implemented
81. Invalid call reference value
88. Incompatible destination
95. Invalid message unspecified
96. Mandatory information element is missing
97. Message type non-existent or not implemented
98. Message not compatible with call state or message type non-existent
or not implemented
99. Information element nonexistent or not implemented
100. Invalid information element contents
101. Message not compatible with call state
102. Recover on timer expiry ? Mandatory IE length error
111. Protocol error, unspecified
127. Interworking, unspecified

[24]

10.3 Generátor DTMF

Následný skript je určený pro vývojové prostředí Matlab a lze pomocí něj generovat zvukový soubor s příponou .wav, který se skládá ze zvolených tónů DTMF volby s definovanou délkou a pauzou mezi nimi.

```
input='1' '2' '3' '4' '5' '6' '7' '8' '9';
fs=8e3;
L=1; %delka tonu
L0=0.08; %delka pauzy
t1=0:1/fs:L-1/fs;
matice=['1' '2' '3' 'A','4' '5' '6' 'B'; '7' '8' '9' 'C'; '*' '0' '#' 'D'];
out=[];
for i=1:length(input)
    [y,x]=find(matice==input(i));
    switch x
        case 1
            f1=1209;
        case 2
            f1=1336;
        case 3
            f1=1477;
        case 4
            f1=1633;
    end
    switch y
        case 1
            f2=697;
        case 2
            f2=770;
        case 3
            f2=852;
        case 4
            f2=941;
    end
    out=[out sin(2*pi*f1*t1)+sin(2*pi*f2*t1) zeros(1,L0*fs)];
end
audiowrite('out_dtmf.wav',0.9*out/max(abs(out)),fs);
soundsc(out)
```

10.4 Vyhodnocovací skript

```
#!/bin/bash

#definice jmena uzivatele
user=ubuntu

#definice cest

hovorycsv=/var/log/asterisk/cdr-custom/hovory.csv
results=/home/$user/vysledky
scripts=/home/$user/scripts
pesq=/home/$user/pesq

#Vyber jednotlivych sloupcu k odpovidajicimu testu z hromadneho
#logu generovaneho Asteriskem

grep "DTMFtest" $hovorycsv | cut -d "," -f '2 4 5 6 7 8 9 11'
    >> $results/dtmftest.csv
grep "Dostupnost" $hovorycsv | cut -d "," -f '2 3 4 5 6 7 11 12'
    >> $results/dostupnost.csv
grep "MOSstest" $hovorycsv | cut -d "," -f '2 4 5 6 7 10 11'
    >> $results/mostest.csv
grep "setuptime" $hovorycsv | cut -d "," -f '2 4 7 11'
    >> $results/setuptime.csv

rm $hovorycsv

#Nacteni hodnoty PESQ a prepocitani na hodnotu MOS pomoci mapovaci funkce

pesq=$(cat $pesq/temp/_pesq_itu_results.txt |
    grep pesq | cut -d " " -f 4)
tmpmos=$(bc -l <<< "0.999+((4.999-0.999)/(1+e((( -1.4945)*($pesq))+4.6607))))")
mos=$(echo "scale=2; $tmpmos/1" | bc)

#Pridani vypoctene hodnoty MOS do logu genreovaneho Asteriskem

cut -d',' -f'1 2 3 4 5 6' $results/mostest.csv > tmpmos2
paste tmpmos2 <(echo "$mos") >> $results/finalmos.csv

#Uklid
```

```

rm $results/mostest.csv
rm $pesq/temp/_pesq_itu_results.txt
rm tmpmos2
unset pesq
unset tmpmos
unset mos

#Testovaci podminka, zda uz dorazila SMS zprava a muze se pokracovat
#ve skriptu

{
if [ ! -f /var/spool/gammu/inbox/*0797990104* ]; then
    echo "\SMS jeste nedorazila!"
    exit 0
fi
}

#testovaci telefonni cislo
#definovano, aby nebyly nacistany sms zpravy z ostatnich cisel
num=797990104

#Nacteni jednotlivych casovych znamek

originated=$(cat /var/spool/gammu/inbox/*$num*)
answered=$(cat $scripts/answered.txt)
recsms=$(cat $scripts/received.txt)

#Vypocet casu a prevod na sekundy

s=$(( $answered - $originated - 150 ))
r=$(( $recsms - $originated - 150 ))
setuptime=$(printf %d.%d ${s%????} ${s##${s%????}})
smsdelivery=$(printf %d.%d ${r%????} ${r##${r%????}})

#Pridani vypoctene hodnoty do logu generovaneho Asteriskem

log=$(cat $results/setuptime.csv)
paste <(echo "$log") <(echo ", $setuptime") >> $results/SETUptime.csv

```

```
#Zapis doby trvani prijmu SMS zpravy do souboru
```

```
echo "$smsdelivery" >> $results/SMStime.txt
```

```
#Uklid
```

```
rm $results/setuptime.csv
```

```
rm $scripts/answered.txt
```

```
rm /var/spool/gammu/inbox/*
```

```
rm $scripts/received.txt
```