



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

MĚŘENÍ KVALITY TELEFONNÍCH HOVORŮ U POBOČKOVÉ ÚSTŘEDNY ASTERISK

MEASURING QUALITY OF TELEPHONE CALLS WITH THE ASTERISK PBX

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. PETR BÍLEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÍT DANĚČEK

BRNO 2011



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Petr Bílek

ID: 78462

Ročník: 2

Akademický rok: 2010/2011

NÁZEV TÉMATU:

Měření kvality telefonních hovorů u pobočkové ústředny Asterisk

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte algoritmy po měření kvality hovorů realizovaných pomocí VoIP technologie. Dále nastudujte možnosti operačního systému Opie, open-source pobočkové ústředny Asterisk a vývojového kitu VOIPAC PXA270M. Následně nastudujte možnosti použití programovacích jazyků C, JAVA a PHP v operačním systému Opie.

Vytvořte návrh koncepce měření kvality hovorů u PBX Asterisk, jež bude realizována na operačním systému Opie ve VOIPAC PXA270M. Koncepce bude koncipovaná jako přehledová pro operátora a bude možné do ní přistupovat i pomocí SSH protokolu. Operátor bude moci volit různé měřicí algoritmy kvality hovoru pro jednotlivé telefonní kanály v PBX Asterisk. Dále mu budou zobrazovány výsledky pro jednotlivé kanály. Výsledky budou také dostupné ve výstupním souboru. Na základě úvodní studie zvolte programovací jazyk, v němž vytvořte zdrojový kód pro vámi navrženou koncepci.

Experimentálně realizujte měření kvality hovorového signálu pro různé algoritmy u jednotlivých telefonních kanálů u PBX Asterisk pomocí vámi vytvořené koncepce.

DOPORUČENÁ LITERATURA:

[1] MEGGELEN, J.V, SMITH, J.Madsen, L. Asterisk™. The Future of Telephony. O'Reilly Media, Inc., Sevastopol 2005. ISBN 0-596-00962-3

[2] RAAKE, A.. Speech Quality of Voip: Assessment and Prediction. John Wiley&Sons New York 2006. ISBN: 978-0-47003-060-8

Termín zadání: 7.2.2011

Termín odevzdání: 26.5.2011

Vedoucí práce: Ing. Vít Daněček

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

ANOTACE

Tato práce obsahuje popis v současné době používaných metod a algoritmů pro měření kvality telefonního hovoru a jejich možné použití v softwarové pobočkové ústředně Asterisk. Dále je vytvořena koncepce systému pro měření kvality telefonního hovoru. Tato koncepce je implementována do embedded systém VOIPAC PXA270M, ve kterém je spuštěna softwarová pobočková ústředna Asterisk. Na základě teoretického rozboru je provedena diskuze mezi obecně rozšířenými programovacími jazyky C, Java a PHP, jejímž výsledkem je výběr vhodného programovacího jazyka pro danou implementaci na embedded systému. Koncepce je realizována pomocí vytvořené aplikaci, napsané ve vybraném programovacím jazyce C. Součástí této práce je i ověření vytvořené aplikace pomocí experimentálních měření na telekomunikačním řetězci. Výsledky experimentálních měření jsou diskutovány v závěrečném zhodnocení.

KLÍČOVÁ SLOVA

Kvalita telefonního hovoru, Mean Opinion Score, R-faktor, softwarová pobočková ústředna Asterisk, vývojový kit VOIPAC.

ABSTRACT

This thesis contains description of the present methods and algorithms for measuring the quality of telephone call on the PBX Asterisk. Further develop the concept of a system for measuring and implementing the call embedded system VOIPAC PXA270M. Based on theoretical analysis is carried out discussions between C, Java and PHP, which results in the selection of appropriate programming language suitable for implementation in embedded system. The concept is realized by creating an application written in a particular programming language. Part of this work is verification created application using experimental measurements of telecommunications system. The results of experimental measurements are discussed in the final evaluation.

KEYWORDS

Quality of telephone call, Mean Opinion Score, R factor, PBX Asterisk, development kit VOIPAC.

BÍLEK, P. *Měření kvality telefonních hovorů u pobočkové ústředny Asterisk: diplomová práce*. Brno: FEKT VUT v Brně, 2011. 48 stran, 2 přílohy. Vedoucí práce Ing. Vít Daněček.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma „Měření kvality telefonních hovorů u pobočkové ústředny Asterisk“ jsem vypracoval samostatně pod vedením vedoucího semestrálního projektu a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedeného semestrálního projektu dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

.....
podpis autora

Poděkování

Děkuji vedoucímu práce ing. Vítu Daněčkovi za velmi užitečnou metodickou pomoc, cenné rady a vedení mé diplomové práce. Dále děkuji ing. Petru Syslovi Ph.D. za pomoc a rady v určitých částech práce.

V Brně dne

.....
podpis autora

OBSAH

Úvod	10
1 Kvalita hovoru ve VOIP a její měření	11
1.1 Kvalita telefonního hovoru	11
1.2 Parametr MOS	11
1.3 Metody vyhodnocování parametru MOS	11
1.4 Subjektivní metody	12
1.5 Objektivní metody	13
1.5.1 Intrusivní metody	13
1.5.2 Neintrusivní metody	16
1.6 Porovnání subjektivních a objektivních metod	18
1.7 E-model	18
2 Vývojový kit VOIPAC PXA270M	24
2.1 Embedded systémy	24
2.2 Základní popis vývojového kitu VOIPAC PXA270M	24
2.3 Operační systém OPIE Linux	25
3 Programovací jazyky pro OS OPIE	26
3.1 Jazyk C	26
3.2 Jazyk Java	26
3.3 Jazyk PHP	27
3.4 Volba vhodného programovacího jazyka pro daný úkol	27
4 Pobočková ústředna Asterisk	28
4.1 Základní charakteristika	28
4.2 Použité součásti PBX Asterisk	28
4.2.1 Konfigurační soubory	28
4.2.2 CLI	29
5 Řešení daného problému	30
5.1 Koncepce systému pro měření kvality telefonního hovoru	30
5.2 Nastavení PBX Asterisk	31
5.3 Experimentální aplikace pro měření kvality telefonního hovoru	32
5.4 Implementace aplikace do vývojového kitu	34
5.5 Ověření funkčnosti a správného návrhu aplikace	35
5.5.1 Volání na klapku pomocí softwarového klienta	35
5.5.2 Volání na klapku pomocí VoIP telefonu Well 3130F	36
5.5.3 Volání mezi dvěma VoIP telefony Well 3130F	39
5.6 Zhodnocení celé koncepce a návrh dalšího postupu	41
6 Závěr	43
Literatura	45
Seznam použitých zkratk	48
Seznam příloh	49

SEZNAM OBRÁZKŮ

Obr. 1.1 Blokové schéma metody PSQM	13
Obr. 1.2 Blokové schéma metody PESQ	15
Obr. 1.3 Průběh mapovací funkce pro přepočítání mezi hodnotami ITU-T P.862 a ITU-T P.862.1	16
Obr. 1.4 Blokové schéma metody 3SQM	17
Obr. 1.5 Nejdůležitější vlivy působící na kvalitu hovoru	18
Obr. 2.1a Celkový pohled na vývojový kit VOIPAC PXA270M s displejem	24
Obr. 2.1b Detailní pohled na modul PXA270M DIMM Max	24
Obr. 2.2a Základní deska s modulem PXA270M DIMM Max	25
Obr. 2.2b Spodní strana základní desky s PCMCIA patičkou a zasunutou CF kartou	25
Obr. 4.1 Příkaz sip show channel v CLI	29
Obr. 5.1 Schéma HW koncepce systému pro měření kvality telefonního hovoru ...	30
Obr. 5.2 Volba hovoru, u kterého má být vypočítána kvalita	33
Obr. 5.3 Výsledný výpočet kvality hovoru	34
Obr. 5.4 Měřicí struktura při volání VoIP telefonem na klapku	36
Obr. 5.5 Uživatelské prostředí emulátoru WANem	36
Obr. 5.6 Naměřená závislost R-faktoru na zpoždění při volání na klapku	37
Obr. 5.7 Naměřená závislost MOS na zpoždění při volání na klapku	38
Obr. 5.8 Referenční závislost R-faktoru na zpoždění signálu	38
Obr. 5.9 Měřicí struktura při volání mezi dvěma VoIP telefony	39
Obr. 5.10 Naměřená závislost R-faktoru na zpoždění při volání mezi účastníky	40
Obr. 5.11 Naměřená závislost MOS na zpoždění při volání mezi účastníky	41
Obr. 5.12 Schéma koncepce systému pro měření kvality hovoru	42

SEZNAM TABULEK

Tab 1.1. Hodnoty stupnice MOS	11
Tab. 1.2 Metody měření a vyhodnocování MOS	10
Tab. 1.3 Porovnání jednotlivých ukazatelů kvality hovoru	19
Tab. 1.4 Hodnoty Ie pro nejpoužívanější kodeky	20
Tab. 1.5 Hodnoty faktoru zvýhodnění A.....	20
Tab. 3.1 Přehled programovacích jazyků vyučovaných na českých VŠ	26
Tab. 5.1 Hodnoty Ie pro podporované hodnoty kodeků v PBX Aterisk	32
Tab. 5.2 Hodnoty parametrů sloužících k výpočtu dle literatury [35]	33
Tab. 5.3 Hodnoty rtt a MOS naměřené při volání na klapku	37
Tab. 5.4 Hodnoty rtt a MOS naměřené při volání mezi dvěma účastníky	40

ÚVOD

S rozvojem datových počítačových sítí, se také rozšiřuje v hromadné míře služba Voice over IP, jež nahrazuje tradiční telefonii. Tato služba poskytuje oproti klasické telefonii řadu výhod, mezi ně patří např. možnost provést telefonní hovor bez nutnosti využít služeb některého telefonního operátora, větší míra implementace (VoIP v mobilních telefonech, v PC), podpora videokonferencí a mnoho dalších. Společně s rozvojem VoIP služeb se rozšířily také softwarové pobočkové ústředny, které v sobě kombinují vlastnosti klasických pobočkových ústřední (PBX), protokolových bran a dalších funkcí. Jejich podíl v malých a středních firmách roste, jelikož jsou nenáročné na údržbu a počáteční investice do vybavení je nižší než u klasických pobočkových ústřední. Služby, které poskytují, jsou ve většině případů postaveny na paketovém provozu, jsou ovšem náchylné ke zpoždění, ztrátám paketů a dalšímu znehodnocení signálu. Tyto veličiny je potřeba měřit a vyhodnocovat a na základě výsledků provádět příslušné úpravy parametrů spojení, tak, aby byla zachována požadovaná kvalita služby.

Tato práce si klade za cíl seznámit čtenáře s problematikou měření a vyhodnocování kvality telefonního hovoru. Čtenář bude seznámen s definováním pojmu kvalita telefonního hovoru a se základním rozdělením metod, kterými je tato kvalita měřena. Toto dělení bude provedeno jak z hlediska celkového přístupu k měření kvality telefonního hovoru, tak i k jeho koncepci a způsobu provedení. Dále v teoretické části budou popsány jednotlivé měřicí metody, které se používají při získání výsledné hodnoty kvality hovoru. Autor při popisu těchto metod vychází zejména z mezinárodních doporučení agentury International Telecommunication Unit s označením ITU-T.

Čtenář práce bude dále seznámen s problematikou embedded systémů, konkrétně vývojového kitu VOIPAC PXA 270M. Tento vývojový kit je zástupcem zařízení vykazující dostatečný výpočetní výkon pro středně složité aplikace při minimálních požadavcích na spotřebu elektrické energie. Softwarová pobočková ústředna Asterisk a její dále používané moduly budou popsány v poslední kapitole teoretické části této diplomové práce.

Praktická část této práce bude věnována popisu koncepce systému pro měření kvality telefonního hovoru, jehož základem bude vývojový kit VOIPAC PXA 270M a softwarová pobočková ústředna Asterisk. Koncepce bude navržena tak, aby PBX Asterisk byl spuštěn na vývojovém kitu. Aplikace měřící kvalitu telefonního hovoru bude vyhodnocovat jednotlivé hovory, které přes PBX Asterisk budou uskutečňovány. Před samotným popisem aplikace bude provedena diskuze, ve které budou zhodnoceny jednotlivé možnosti nejčastěji používaných programovacích jazyků a v závislosti na složitosti implementaci na vývojový kit VOIPAC PXA 270M bude vybrán nejvhodnější z nich.

V praktické části bude dále popsán způsob ověření správného návrhu a funkce vytvořené aplikace měřící kvalitu telefonního hovoru. Experimentálním měřením bude ověřena celá koncepce včetně autorem vytvořené aplikace. Toto experimentální měření bude realizováno pomocí dostupných síťových nástrojů, jež umožňují simulovat i analyzovat provoz na zatížené datové síti obsahující mj. i právě probíhající telefonní hovory. Výsledky experimentálních měření budou v závěru zhodnoceny.

1 KVALITA HOVORU VE VOIPA JEJÍ MĚŘENÍ

1.1 Kvalita telefonního hovoru

Tento termín je především spjatý s digitálním přenosem hlasu přes datovou síť, která je postavená zejména na komunikačních protokolech TCP, UDP a IP. Je jedním z nejdůležitějších parametrů při hodnocení hovoru uskutečňovaném v rámci této sítě. Kvalita hovoru je velmi subjektivním hodnocením řečového signálu, který je ovlivněn například vlivem okolí, způsobem mluvy řečníka, a také samotným přenosovým kanálem. Měření kvality hovoru je definováno v doporučení ITU-T P.10 [9]. Nejvíce používanou metodou k měření kvality hovoru je MOS (viz dále).

1.2 Parametr MOS

Zkratka MOS znamená Mean Opinion Score a je definována jako hodnota ze stupnice, která byla stanovena statisticky na základě subjektivního hodnocení testovaného vzorku populace. Tato stupnice je pětibodová, jak je vidět v tab. 1.1.

Tab. 1.1: Hodnoty stupnice MOS.

Hodnota MOS	Kvalita	Popis kvality
5	Vynikající (Excellent)	Bez znatelného rušení
4	Dobrá (Good)	Znatelné rušení, které neobtěžuje
3	Průměrná (Fair)	Znatelné rušení, které obtěžuje
2	Špatná (Poor)	Rušení velmi obtěžující, porozumění řeči je obtížné
1	Mizerná (Bad)	Rušení velmi obtěžující, řeči nelze porozumět

Existuje několik druhů stupnic, zde uvedená je nejčastěji používanou a nazývá se Stupnicí poslechové kvality. Dalšími jsou například Stupnice poslechového úsilí (MOS_{LE}) nebo Stupnice preference hlasitosti (MOS_{LP}). Tato problematika je podrobně popsána v [9, 25].

1.3 Metody vyhodnocování parametru MOS

Hodnotu MOS lze získat několika způsoby, nejčastěji se používají tři metody: subjektivní, objektivní a odhadované.

V případě subjektivních metod se používá hodnocení od konkrétního vzorku posluchačů na základě jejich subjektivního vnímání daného hovoru. Jde ovšem o velmi časově i finančně náročné metody, jelikož je potřeba velkého vzorku posluchačů, aby bylo hodnocení co nejvíce statisticky přesné. Každý testovaný posluchač se může zúčastnit testu pouze jednou. Dále je vyžadováno, aby byl v této oblasti laik. Pokud tyto podmínky nejsou dodrženy, dochází ke zkreslení celého měření. I přes uvedená omezení jde o nejpřesnější metody měření a vyhodnocení MOS.

Objektivní měřicí metody byly zavedeny právě z důvodu náročnosti metod subjektivních. Není již potřeba posluchačů, výsledné hodnoty se počítají pomocí navržených algoritmů, které jsou implementovány a vyhodnocovány výpočetní technikou. Jejich hlavní předností je výpočet MOS v reálném čase. Výsledky, které jsou pomocí nich dosahované, však nejsou tak přesné, jako ty, které se dosahují pomocí měření subjektivními metodami. Využívají se všude tam, kde by nasazení subjektivních metod bylo nákladné, případně pro sledování průběhu kvality řeči v reálném čase, například u operátorského pracoviště.

V odhadovaných metodách se využívá „popisu daného systému a empirických hodnot kvality, které jsou daným parametrům vlastní. Takovýto způsob nezohledňuje dynamické jevy, které mohou v systému nastat a proto se tento způsob používá zejména k přibližnému popisu a tedy k odhadu kvality při návrhu nějakého nového systému.“ Převzato z [25].

Dalším způsobem, jak lze MOS vyhodnocovat, je způsob koncepce celého testu. Celé měření můžeme připravit jako poslechové nebo konverzační. Při poslechovém testu nemusíme mít celý přenosový kanál. Stačí pouze, aby byly testujícím subjektům pouštěny předem připravené nahrávky, které pak hodnotí. U konverzačního testu je již vyžadována dvojice subjektů na konci telekomunikačního řetězce, přes který probíhá testovaný hovor.

Pokud zkombinujeme základní tři způsoby měření MOS a dvě uvedené koncepce měření dostaneme celkem 6 možností, jak můžeme MOS měřit a vyhodnocovat. Přehled se nachází v tab. 1.2, která je definována v doporučení ITU-T P.800.1 [11].

Tab. 1.2: Metody měření a vyhodnocování MOS.

Metoda/Koncepce testu	Poslechový	Konverzační
Subjektivní	MOS-LQS	MOS-CQS
Objektivní	MOS-LQO	MOS-CQO
Odhadovaný	MOS-LQE	MOS-CQE

Jednotlivé metody si nejsou rovny, obecně platí, že subjektivní metody jsou přesnější, než zbývající dva způsoby. Při jejich porovnávání je proto potřeba vždy učinit přepočty, který je pro každý převod jiný (bude uvedeno na příkladech dále).

1.4 Subjektivní metody

Jak bylo výše uvedeno, jde o metody, které jsou založeny na hodnocení živých posluchačů. Jde o nejpresnější metody k zjišťování kvality hovoru. Používá se pro ně následující vyhodnocování:

- ACR – Absolute Category Rating – Každý z testujících subjektů přiřadí přímou hodnotu MOS danému poslouchanému vzorku nebo hovoru. Metoda je používána jak pro analogový, tak pro digitální přenos řeči.
- Quantal-Response Detectability Tests – Jedná se o metodu, která je nevhodnější pro vyhodnocování prahových úrovní a jejich pravděpodobností v signálu.
- DCR – Degradation Category Rating – Metoda, která je alternativní k ACR. Porovnává se zde původní nedegradovaný signál se signálem, který prošel telekomunikačním řetězcem. Následně se vyhodnotí míra degradace a pomocí ní se určí hodnota MOS. Hodí se zejména pro optimalizaci systému, kdy oba signály jsou již kvalitní.
- CCR – Comparison Category Rating – Stejně jako v DCR se porovnávají dva signály, které jsou vysoké kvality. Metoda se hodí zvláště pro systémy, které upravují kvalitu řeči na vstupu, například při potlačení ozvěny.
- The threshold method – Další z metod hodící se pro optimalizaci systému. V tomto případě se porovnávají rovnou celé systémy. Jeden z nich je referenční, druhý testovaný a hodnotí se, který signál je lepší.

Přesnější popis subjektivních metod pro získání hodnocení MOS lze nalézt v doporučení ITU-T P.800 [11].

1.5 Objektivní metody

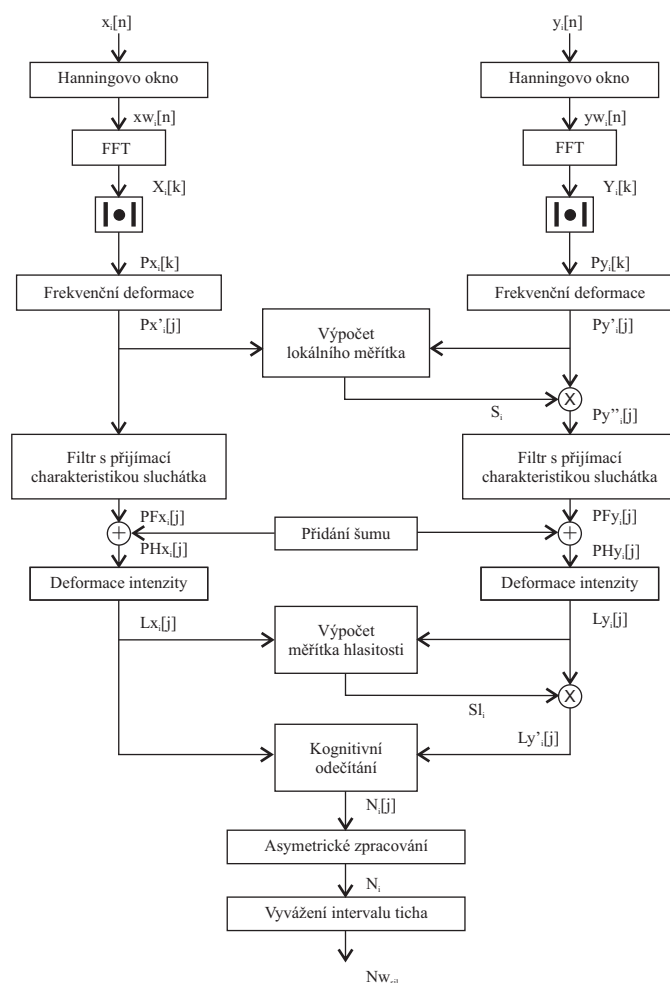
Objektivní metody jsou založeny na různých algoritmech, které se snaží přiblížit hodnotám získaným ze subjektivního měření signálu. Jejich hlavní výhodou je zpracování v reálném čase. Objektivní metody se dají rozdělit na dva druhy – intrusivní a neintrusivní.

1.5.1 Intrusivní metody

Tyto metody jsou založeny na srovnání původního signálu se signálem, který byl přenesen pomocí telekomunikačního řetězce. Jejich výsledky se nejvíce blíží subjektivním metodám vyhodnocování MOS. Využívají metod psychoakustického modelu smyslového vnímání člověka [27]. U těchto metod je největší problém získání originálního signálu, metody nelze použít, pokud k němu nemáme přístup. Pokud ovšem originální signál získáme, můžeme měřit zpoždění a pozorovat degradaci signálu při přenosu. Pomocí těchto metod dosahujeme hodnocení MOS-LQO. Mezi intrusivní metody patří například PSQM, PAMS nebo PESQ [11, 27].

Metoda PSQM

Objektivní měřicí metoda PSQM (Perceptual Speech Quality Measure) byla vymyšlena roku 1993. Vychází z obecnější metody PAQM. Popis celé metody lze nalézt v doporučení ITU-T P.861 [12], které vzniklo v roce 1996. Blokové schéma metody ukazuje obr. 1.1



Obr. 1.1: Blokové schéma metody PSQM.

Do systému vstupují vzorky původního signálu $x_i[n]$ a signálu degradovaného $y_i[n]$, které jsou předzpracovány a transformovány do frekvenční oblasti. Lineární frekvenční měřítko je poté v bloku frekvenční deformace přepočítáno na měřítko výšky tónu. Následně jsou oba signály filtrovány. Je použita přenosová charakteristika přijímacího zařízení (např. sluchátko, reproduktor, mikrofon). Poté se přidá šum, který simuluje vjemy, jenž se mohou vyskytnout na pozadí hovoru (dnes se používá simulace kancelářského prostředí), a tím modelovat reálné prostředí. Blok deformace intenzity slouží k reprezentaci komprimované hlasitosti v závislosti na výšce tónu a čase. U deformovaného signálu se ještě přepočítá měřítko hlasitosti. Následně se oba signály od sebe odečtou a je odvozena zvuková chyba, která je stále funkcí výšky tónu a času. Poslední dva bloky se vyrovnávají s problémy, které by do výsledného ukazatele MOS přinesly použité kodeky a příliš velké intervaly ticha v hovoru (velké intervaly ticha zkreslují daný hovor, protože jsou z velké části tvořeny šumem). Výsledek celé operace se pak ještě přepočítá pomocí vzorce (1.2). Tento výsledek by měl predikovat hodnotu MOS, kterou bychom dostali při použití subjektivní metody ACR. Více o PSQM lze nalézt v [12, 22, 26].

Metoda PSQM+

Jelikož byl algoritmus PSQM navrhován pro měření převážně na analogových linkách, má určité nedostatky. Především jde o problémy při nasazení v digitálních linkách a vypořádání se se ztrátovostí paketů. Časové zarovnávání není úplně přesné, zvláště je problematické na zarušených linkách. „Standardizované časové zarovnání je založené na detekci počátečního 0,5ms intervalu okna, kde referenční a testovaný signál dosahují určitou minimální energii. Tento úsek je bráný jako startovací úsek obou vzorků. Jakýkoli šum, který se objeví během měření, může způsobit detekci, a to vede k celkovému nesprávnému výsledku.” Přeloženo z [26]. Pomocí algoritmu časového zarovnání (zaveden firmou OPTICOM) se tato změna dá eliminovat. Dále je zde oproti klasickému PSQM změněn blok asymetrického zpracování a jsou použity škálovací algoritmy pro vyrovnání ztrátovosti paketů. Přesto metoda vykazuje problémy při měnícím se zpoždění paketů, tudíž pro měření MOS ve VoIP není tou nejvhodnější [26].

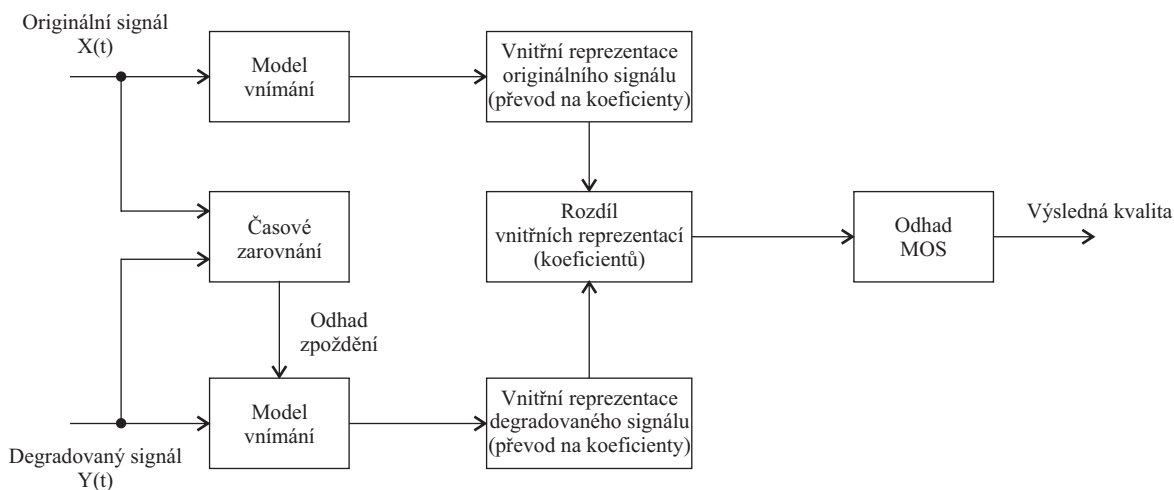
Metoda PAMS

Metoda, která byla navržena firmou British Telecom. Problém použití PSQM ve VoIP spočívá v tom, že nebyl navržen pro paketový přenos dat. Algoritmus není uzpůsoben na zpoždění jednotlivých vzorků. V PAMS je časová a amplitudová korelace prováděna v dílčích krocích, ne jako u PSQM na celém záznamu najednou. Po převedení do frekvenční oblasti je vytvořena poslechová plocha, kde jsou jednotlivé vzorky umístěny (jde defakto o dva vektory). Rozdíl mezi hodnotami na stejném místě v obou vektorech představuje chybovou plochu, která se používá pro odhad hodnotících parametrů. Z těchto parametrů pak můžeme zjistit výsledné MOS [6].

Metoda PESQ

Metoda PESQ – Perceptual Evaluation of Speech Quality – je definována dle doporučení ITU-T P.862. I zde se porovnává originální signál se signálem degradovaným po přenosu telekomunikačním řetězcem. PESQ je založen na psychoakustickém modelu a hodnocení vzorků z PSQM+ a metodě rekurentní časové korelace z PAMS, která zajišťuje zpracování

příchozích vzorků v krocích. Blokové schéma PESQ ukazuje obr. 1.2. Nejprve se vstupní signály $X(t)$ a $Y(t)$ zarovnají v čase a vypočte se hodnota zpoždění mezi nimi. Pak se oba signály převedou na interní reprezentaci. Tato reprezentace se blíží psychofyzické reprezentaci zvukového signálu ve sluchovém ústrojí člověka. Bere se zde ohled na vnímání frekvence a hlasitosti. To je dosaženo v několika krocích – časovou synchronizací, přizpůsobením úrovní jedné z kalibrovaných poslechových úrovní, časově–frekvenční mapování, frekvenční deformací a kompresí měřítko hlasitosti. Interní reprezentace je pak zpracována pomocí psychoakustické transformace. Ta parametry obou signálů zpracovává matematickým modelem sluchového ústrojí člověka. Výsledkem jsou matice, jež stejně jako v PAMS představují vektory zpracovaných vzorků obou signálů. Příslušné prvky se opět odečtou. Pokud se objeví úsek s výrazně vyšší chybou, je znovu časově zarovnán. Sečtou se odděleně kladné a záporné rozdíly mezi maticemi a výsledky jsou vynásobeny různými koeficienty (vyšší význam šumu než chybějících úseků, tato skutečnost vychází z vlastností lidského ucha). Váňované sumy se odečtou od maximálního koeficientu 5 a dostáváme hodnotu MOS pro daný vzorek signálu.



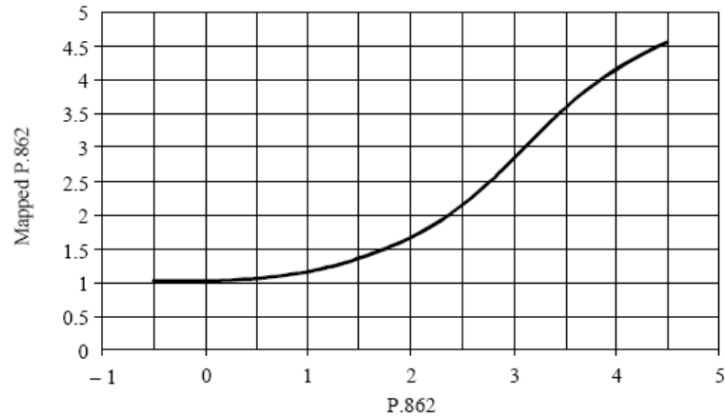
Obr. 1.2: Blokové schéma metody PESQ.

Výsledné hodnocení MOS se v případě metody PESQ pohybuje v rozmezí $-0,5-4,5$. Nastává tedy problém při srovnávání a přepočtu na subjektivní hodnotu MOS, která je v rozsahu $1-5$. Z tohoto důvodu bylo rozšířeno doporučení ITU-T P.682 a vzniklo ITU-T P.682.1. Mapovací funkce je popsána rovnicí (1.1) a její průběh lze vidět na obr. 1.3. Převzato z [13].

$$y = 0,999 + \frac{4,999 - 0,999}{1 + e^{-1.4945 * 4.6607}} \quad (1.1)$$

Metoda PESQ dosahuje kvalitativních výsledků zejména tam, kde vstupním signálem je lidská řeč a kde může docházet ke zpožděním a chybám v přenosovém kanálu, případně pokud je signál v přenosové cestě zašuměný [6, 13, 14, 18, 25, 26, 27, 32].

Existuje celá řada dalších intusivních metod (například PEAQ, MNB nebo EMBSD). Jejich výčet a popis však přesahuje rámec této diplomové práce.



Obr. 1.3: Průběh mapovací funkce pro přepočítání mezi hodnotami ITU-T P.862 a ITU-T P.862.1.

1.5.2 Neintrusivní metody

Druhou skupinou objektivních metod jsou metody, u nichž se používá k výpočtu hodnoty MOS pouze signál, který prošel telekomunikačním řetězcem. Není potřeba znát signál původní. Z tohoto důvodu se neintrusivní metody více hodí na monitorování kvality hovoru v síti, a to i v reálném čase. Nevýhodou těchto metod je menší přenos v porovnání s metodami intrusivními, případně subjektivními. Hodnocení MOS získané těmito metodami je pouze odhadované MOS-LQE (neplatí vždy, viz dále). Intrusivních metod je celá řada, například 3SQM, INMD, CCI, LCQA nebo ANIQUE [25].

Metoda 3SQM

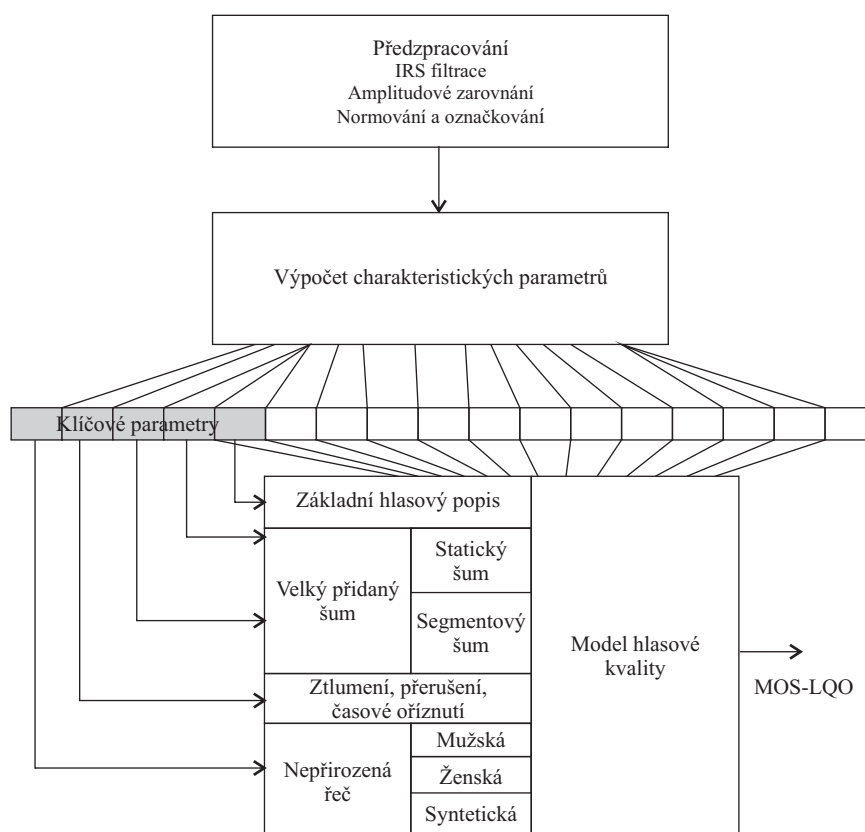
Nejznámější neintrusivní metodou je metoda Single Sided Speech Quality Measure, zkráceně 3SQM. Jako doporučení ITU-T P.563 byla vydána v roce 2004. Jde tedy o metodu, která je o něco mladší, než popisované intruzivní metody. Při její aplikaci jsou vzaty v úvahu různé deformace, které mohou přenesený signál postihnout. Výsledky této metody predikují hodnoty, které lze zařadit do stupnice MOS-LQO, tudíž stejné jako intrusivní metody. Přesto má jednu nevýhodu. Tou je omezení použití pouze na úzkopásmové telefonní aplikace [18]. Blokové schéma metody je vidět na obr. 1.4.

Signál je nejdříve předzpracován pomocí modelu klasického mikrotelefonu. Značí to filtraci, amplitudové zarovnání a normování signálu na hodnotu -26 dBov. V dalším bloku se rozpoznává, v které části hovoru jsou aktivní informace (samotný hovor). Tyto části jsou označovány a dále jsou zpracovávány. Následuje několik analýz, které jednotlivé části rozdělí podle charakterizujících parametrů. Podle klíčových parametrů signálu je určena hlavní třída rušení. Tyto třídy jsou celkem tři, každá má vymezen jeden blok zpracování:

- Vysoký přídavný šum – Pokud je jako hlavní třída určen vysoký přídavný šum, rozlišuje se, zda jde o šum statický, který je rozložen rovnoměrně v celém signálu, či zda jde o šum segmentový, který koreluje s výkonovou obálkou signálu. Segmentový šum se poté ještě dělí na globální (šum v pauzách mezi větami) a lokální (mezi hláskami).
- Ztlumení, přerušení a časové oříznutí – V tomto bloku se vyhodnocují nepřírozená ticha v signálu, případně jeho nenadálé přerušení ať už chvilkové, nebo definitivní.

3SQM je schopno poznat tyto anomálie v hovoru od standardního ukončení slova nebo věty. Děje se tak pomocí detektoru řečové aktivity.

- Nepřirozená řeč – Třetí blok je založen na LPC modelu lidského hlasového ústrojí. Jde o soustavu generátorů a filtrů, které se snaží kopírovat lidskou řeč, která přijde na vstup, a podle toho se určuje odpovídající degradace signálu (jak moc je signál podobný lidskému hlasu). Tato analýza se provádí zvlášť pro mužský a zvlášť pro ženský hlas. Pokud se v signálu vyskytuje velké množství vysoko-frekvenčních složek, je signál posuzován jako syntetický.



Obr. 1.4: Blokové schéma metody 3SQM.

Z těchto tří tříd se pak vypočte 39 koeficientů signálu. Použije se z nich 11 a podle toho, která třída zkreslení dominuje, se vybere dalších 12. Ty se mohou s těmi 11 stálými překrývat. Následně se z koeficientů vypočte lineární kombinace, která představuje predikovanou hodnotu MOS [10, 18, 24, 27].

Metody INMD a CCI

Doporučení ITU-T P.561 – In-Service Non-intrusive Measurement Device popisuje parametry, které je potřeba vyhodnotit v signálu, který prošel telekomunikačním řetězcem. Je to například úroveň signálu, SNR, řečová aktivita nebo echo. V doporučení ITU-T P.562 – Call Clarity Index je popsáno, jak z těchto parametrů vypočítat jediný výsledný parametr, kterým je predikovaná hodnota MOS [6].

1.6 Porovnání subjektivních a objektivních metod

Z předchozího textu vyplývá, že subjektivní metody mají větší vypovídající hodnotu, než ty objektivní. Jsou ovšem mnohem nákladnější, takže se v praxi spíše používají metody objektivní. Je potřeba mezi těmito dvěma druhy metod provést korelaci, aby výsledky z objektivního testování byly srovnány s výsledky hodnocení subjektivního. Děje se tak pomocí následujícího Pearsonova vztahu (1.2):

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}, \quad (1.2)$$

kde r – korelační koeficient,

x_i – MOS hodnota pro konkrétní i -tý vzorek subjektivní metody,

\bar{x} – aritmetický průměr všech hodnot subjektivní metody,

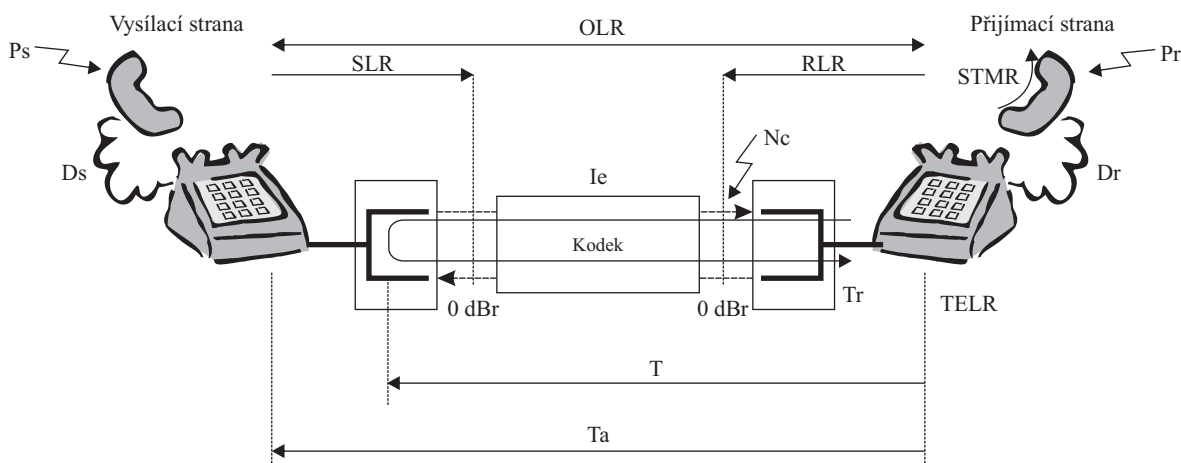
y_i – MOS hodnota pro konkrétní i -tý vzorek objektivní metody,

\bar{y} – aritmetický průměr všech hodnot objektivní metody.

Korelační koeficient nabývá hodnot v intervalu $<0; 1>$. Převzato z [18].

1.7 E-model

K měření kvality hovoru se nevyužívají jen subjektivní a objektivní metody. Existuje doporučení ITU-T G.107 (rozšířeno v G.108), kde je definován tzv. E-model. Byl navržen zejména pro plánování přenosových systémů, kde je potřeba při návrhu vzít v potaz mnoho parametrů, které působí na kvalitu hovoru. Model se snaží vypočítat výslednou kvalitu hovoru v závislosti na kombinaci všech parametrů. První verze doporučení vypracoval v roce 2000 Švéd Nils-Olof Johannessen, poslední revize a doplnění se uskutečnilo v roce 2008. Základní schéma nejdůležitějších parametrů lze nalézt na obr. 1.5.



Obr. 1.5: Nejdůležitější vlivy působící na kvalitu hovoru.

Význam zkratk parametrů je následující:

D_s, D_r – hodnota D pro telefon na vysílací nebo přijímací straně,

I_e – faktor zhoršení vlivem nízkorychlostních kodeků,

N_c – úroveň šumu vztažená k místu relativní úrovně 0,

OLR – celková míra hlasitosti,

P_s, P_r – hluk v místnosti na vysílací nebo přijímací straně,

RLR – míra hlasitosti na přijímací straně,

SLR – míra hlasitosti na vysílací straně,

T – zpoždění v jednom směru,

T_a – celkové zpoždění v jednom směru,

T_r – zpoždění smyčky,

$TELR$ – míra hlasitosti ozvěny na straně hovořícího.

Převzato z [18, 24].

Výstupem E-modelu je hodnota, která mnohem více odpovídá kvalitě hovoru v moderních sítích, ať už úzkopásmových nebo širokopásmových. Nazývá se R-faktor a dosahuje hodnot v rozmezí 0 až 100. Jeden z důvodů, proč byl zaveden E-model, je přechod na plně digitální linky, kde již není potřeba definovat vztažný útlum (analogový signál je již jen v koncovém zařízení). Zavedl se proto parametr míra hlasitosti. E-model může mít kromě výstupu v podobě R-faktoru, ještě tzv. hodnoty procentuální – GoB (Good or Better) a PoW (Poor or Worse), případně vyjádření pomocí Gaussovské chybové funkce [7]. Odpovídající parametry MOS, R-faktoru, GoB a WoS lze nalézt v tab. 1.3. Lze z ní vyzorovat, že vzhledem k MOS je R-faktor definován od hodnoty 50 výše. Sítě, které mají R-faktor nižší jak tuto hodnotu, se nevyplatí provozovat, z důvodu velmi špatné kvality hovoru.

Tab. 1.3: Porovnání jednotlivých ukazatelů kvality hovoru.

R faktor [-]	MOS [-]	GoB [%]	PoW [%]
100–90	4,5–4,34	100–97	0
90–80	4,34–4,03	97–89	0
80–70	4,03–3,60	89–73	0–6
70–60	3,60–3,10	73–50	6–17
60–50	3,10–2,58	50–27	17–38

Výsledná hodnota R-faktoru je dána součtem několika parametrů, které postihují vlastnosti daného přenosového systému. Výsledný vztah je tento (1.3) [7],

$$R = R_o - I_s - I_d - I_e + A, \quad (1.3)$$

kde

R – výsledná hodnota R-faktoru,

R_o – základní hodnota kvality odvozená z poměru signál/šum,

I_s – faktor znehodnocení vlivem lineárního zkreslení (postihuje všechny degradace signálu,

kteře nastávají během přenosu, např. pokles úrovně signálu, šum, vliv zpětné vazby),

I_d – faktor znehodnocení vlivem zpoždění (zde je zahrnuto echo na blízkém a vzdáleném konci),

I_e – faktor znehodnocení vlivem použití nízkorychlostních kodeků (hodnoty I_e pro nejpoužívanější kodeky lze nalézt v tab. 1.4. Převzato z [7, 24]),

A – faktor očekávání, kde se zohledňuje mobilita koncového zařízení, viz tab. 1.5. Převzato z [7].

Tab. 1.4: Hodnoty I_e pro nejpoužívanější kodeky.

Kodek	I_e [-]
PCM G.711 (alaw, ulaw)	0
ADPCM G.726	7
LD-CELP G.728	7
CS-ACELP G.729	10
MP-MLQ G.723.1	15
ACELP G.723.1	19
GSM Full-rate RPE-LTP	20
GSM Half-rate VSELP	23

Tab. 1.5: Hodnoty faktoru zvýhodnění A .

Typ koncového zařízení	A [-]
Pevný terminál	0
Přenosný DECT	5
Mobilní GSM	10
Satelitní terminál	20

Tyto základní parametry se dále počítají podle následujících vztahů:

Poměr signálu k šumu R_o :

$$R_o = 15 - 1,5(SLR + N_o), \quad (1.4)$$

kde N_o značí součet výkonů z různých šumových zdrojů a počítá se:

$$N_o = 10 \log \left(10^{\frac{N_c}{10}} + 10^{\frac{N_{os}}{10}} + 10^{\frac{N_{or}}{10}} + 10^{\frac{N_{fo}}{10}} \right), \quad (1.5)$$

kde N_c je suma všech okruhů výkonů šumu vztažených k místu s hodnotou 0 dBr,

N_{os} je hodnota ekvivalentního okruhu výkonu šumu vztaženého k místu s hodnotou 0 dBr, způsobeného šumem v místnosti na vysílací straně P_s a lze ho získat ze vztahu:

$$N_{os} = P_s - SLR - D_s - 100 + 0,004(P_s - OLR - D_s - 14)^2. \quad (1.6)$$

Jeho protihodnotou je parametr N_{or} , který značí ekvivalentní okruh výkonu šumu vztaženého k místu s hodnotou 0 dBr způsobeného šumem v místnosti na přijímací straně P_r . Počítá se následovně:

$$N_{or} = RLR - 121 - P_r + 0,008(P_r - 35)^2. \quad (1.7)$$

Parametr Pre se nazývá efektivní šum místnosti a je způsobován posílením parametru Pr na straně volaného účastníka. Vztah pro jeho výpočet je následovný:

$$Pre = Pr + 10 \log \left(1 + 10^{\frac{(10-LSTR)}{10}} \right). \quad (1.8)$$

Posledním parametrem ovlivňujícím No je šum, který se odráží od podlahy na přijímací straně, značí se Nfo a počítá se:

$$Nfo = Nfor + RLR. \quad (1.9)$$

Hodnota $Nfor$ udává úroveň šumu na pozadí na přijímací straně.

Faktor degradace signálu vlivem lineárního zkreslení se počítá následovně:

$$Is = Iolr + Ist + Iq. \quad (1.10)$$

První člen rovnice $Iolr$ udává pokles kvality zapříčiněný příliš nízkou hodnotou celkové hlasitosti a počítá se:

$$Iolr = 20 \left[\left\{ 1 + \left(\frac{Xolr}{8} \right)^8 \right\}^{\frac{1}{8}} - \frac{Xolr}{8} \right], \quad (1.11)$$

kde

$$Xolr = OLR + 0,2(64 + No - RLR). \quad (1.12)$$

Druhý člen rovnice (1.10) Ist zohledňuje zhoršení kvality hovoru vlivem vznikajících ozvěn a vyjádří následujícím vztahem:

$$Ist = 12 \left[1 + \left(\frac{STMRo - 13}{6} \right)^8 \right]^{\frac{1}{8}} - 28 \left[1 + \left(\frac{STMRo + 1}{19,4} \right)^{35} \right]^{\frac{1}{35}} - 13 \left[1 + \left(\frac{STMRo - 3}{33} \right)^{13} \right]^{\frac{1}{13}} + 29, \quad (1.13)$$

kde

$$STMRo = -10 \log \left(10^{\frac{STM}{10}} + e^{\frac{-T}{4}} 10^{\frac{TELR}{10}} \right), \quad (1.14)$$

STM v této rovnici představuje míru potlačení vlastního hovoru.

Poslední člen rovnice (1.10) Iq představuje zhoršení kvůli kvantizačnímu zkreslení a je vyjádřen:

$$Iq = 15 \log(1 + 10^Y + 10^Z), \quad (1.15)$$

$$Y = \frac{Ro - 100}{15} + \frac{46}{8,4} - \frac{G}{9}, \quad (1.16)$$

$$Z = \frac{46}{30} - \frac{G}{40}, \quad (1.17)$$

$$G = 1,07 + 0,258Q + 0,0602Q^2, \quad (1.18)$$

$$Q = 37 - 15 \log(qdu), \quad (1.19)$$

kde

veličina qdu v rovnici (1.19) je jednotkou kvantizačního zkreslení a jde o hodnotu, která je společná pro celé spojení.

Třetím členem vztahu (1.3) je Id , který reprezentuje zhoršení vlivem zpoždění signálu procházejícího telekomunikačním řetězcem. Jeho výpočet je dán rovnicí:

$$Id = Idte + Idle + Idd. \quad (1.20)$$

$Idte$ vyjadřuje echo na straně hovořícího (rovnice (1.21)), $Idle$ pak na straně naslouchajícího (vztah (1.25)). Idd reprezentuje vliv absolutního zpoždění Ta (rovnice (1.27)). Jejich výpočet je následující:

$$Idte = \left[\frac{Roe - Re}{2} + \sqrt{\frac{(Roe - Re)^2}{4} + 100} - 1 \right] (1 - e^{-T}), \quad (1.21)$$

$$Roe = -1,5(No - RLR), \quad (1.22)$$

$$Re = 80 + 2,5(TErv - 14) \quad (1.23)$$

$$TErv = TELR - 40 \log \left(\frac{1 + \frac{T}{10}}{1 + \frac{T}{150}} \right) + 6e^{-0,3T^2}, \quad (1.24)$$

$$Idle = \frac{Ro - Rle}{2} + \sqrt{\frac{(Ro - Rle)^2}{4} + 169}, \quad (1.25)$$

$$Rle = 10,5(WEPL + 7)(Tr + 1)^{0,25}. \quad (1.26)$$

Pro absolutní zpoždění $Ta < 100$ ms platí

$$Idd = 0,$$

pro $Ta > 100$ ms

$$Idd = 25 \left\{ (1 + X^6)^{\frac{1}{6}} - 3 \left[1 + \left(\frac{X}{3} \right)^6 \right]^{\frac{1}{6}} + 2 \right\}, \quad (1.27)$$

kde

$$X = \frac{\log\left(\frac{Ta}{100}\right)}{\log 2}. \quad (1.28)$$

Detailní vysvětlení všech uvedených pojmů a vztahů lze nalézt v doporučení ITU-T G.107, odkud jsou převzány [7].

Přestože bylo řečeno, že stupnice R-faktoru může nahradit stupnici MOS, existuje mezi nimi převodní vztah (1.29) a převodní charakteristika, která je k nalezení v Příloze 1. Charakteristika je definována v doporučení ITU-T G.107 [7].

$$MOS = \begin{cases} R < 0 \Rightarrow MOS = 1 \\ 0 \leq R < 100 \Rightarrow MOS = 1 + 0,035 \cdot R + R \cdot (R - 60) \cdot (100 - R) \cdot 7 \cdot 10^{-6} \\ R \geq 100 \Rightarrow MOS = 4,5 \end{cases} \quad (1.29)$$

Měření R-faktor, tedy vhodně doplňuje jak subjektivní, tak i objektivní metody měření kvality hovoru. Záleží na dané aplikaci a konkrétním problému, abychom rozhodli, který postup získání kvality hovoru použijeme. Obecně však platí, že nejlepší výsledky vždy dávají metody subjektivní.

2 VÝVOJOVÝ KIT VOIPAC PXA270M

2.1 Embedded systémy

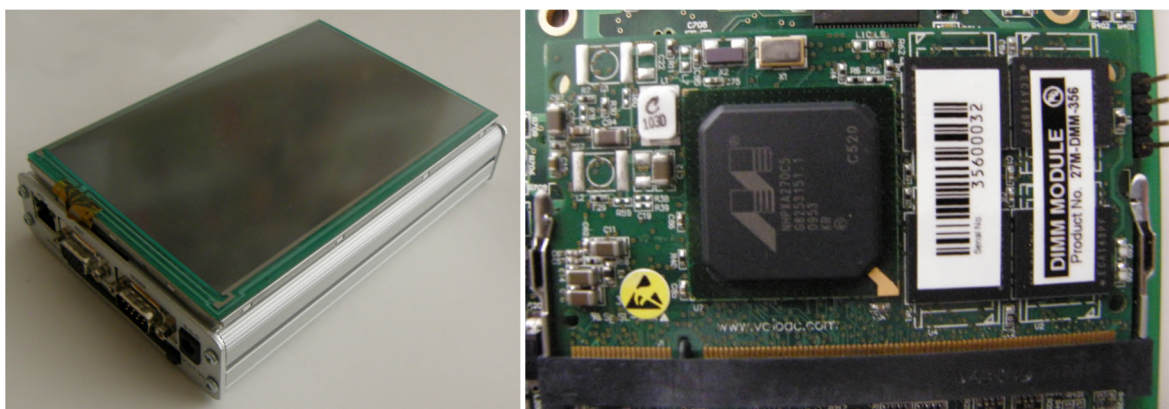
Překlad termínu embedded system je nejednoznačný, můžeme nalézt výrazy jako „vestavěný systém“ nebo „zabudovaný systém“. Nejčastěji se termín nepřekládá a používá se spojení embedded systém. Význam pojmu je taktéž velmi obtížně definovatelný. Podle [4] se jedná o systém, který je založen na mikroprocesorovém základě. Jeho hlavním cílem je řízení specifických funkcí, do kterých pak koncový uživatel již minimálně zasahuje (nemůže již změnit dané funkce přidáním nebo nahrazením softwaru, vyjma upgradu firmwaru). Samotný procesor embedded systému může být čtyřbitový mikrokontroler, nebo také 128bitový DSP procesor. Taktéž se objevují v široké škále způsobu řešení softwaru, od firmwaru nahraném v paměti ROM až po operační systémy Linux, Windows aj. Z předchozích vět plyne, že jejich zaměření je velmi široké, od hraček, přes řízení aplikací v automatizaci, využití v mobilních komunikacích, až po kontrolu určitých procesů v jaderných elektrárnách [4, 19].

2.2 Základní popis vývojového kitu VOIPAC PXA270M

Jde o zařízení, které je navrženo jako plnohodnotné embedded PC a vývojový kit. Jeho určení je hlavně pro aplikace automatizační, síťové, terminálové a jiné, které potřebují operační systém, ale jsou určeny pro úzce zaměřené použití (nepotřebují OS se všemi jeho funkcemi).

Základem je modul PXA270M DIMM Max, jehož hlavními atributy jsou:

- CPU PXA270M (520 MHz, defaultně nastaveno 312 MHz),
- Flash NAND paměť 256 MB (16 bit),
- SDRAM paměť 128 MB (32 bit),
- předinstalovaný Linux 2.6.

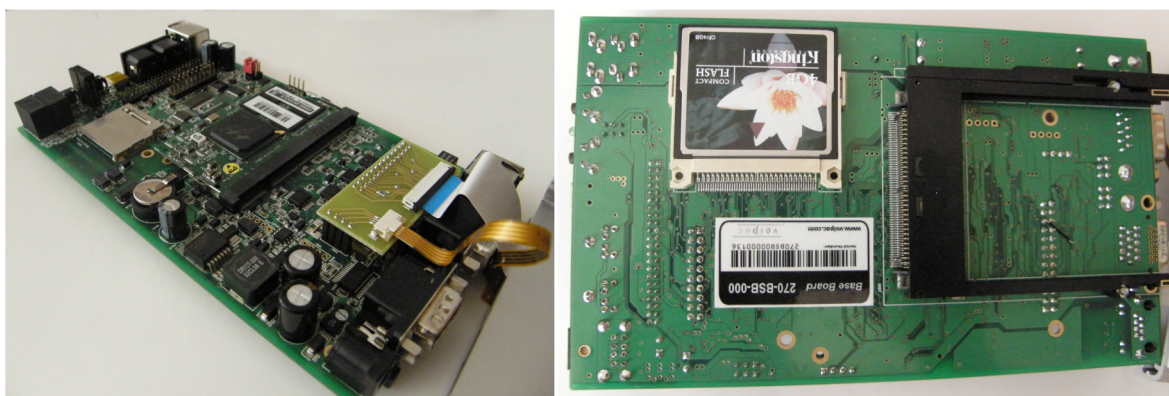


Obr. 2.1: a) Celkový pohled na vývojový kit VOIPAC PXA270M s displejem.
b) Detailní pohled na modul PXA270M DIMM Max.

DIMM Modul je osazen na základní desce, která má tyto vlastnosti:

- Napájení adaptérem 7–37 V (ideální nad 9 V),
- 1 × konektor RJ45 (10/100 Mb/s Ethernet),
- 1 × konektor VGA,
- 1 × konektor Rs232,
- 1 × konektor USB Mini,

2 × konektor USB,
2 × konektor PS/2 (pro klávesnici a myš),
1 × konektor JTAG (umístěn na základní desce),
1 × socket PCMCIA,
1 × socket Compact Flash (osazen 4GB kartou),
1 × socket MMC/SD (osazen 2MB kartou),
TFT displej, rozlišení $800 \times 3 \times 480$ bodů, velikost 7 palců, napájení 5 V,
další specifikace viz literatura [41].



Obr. 2.2: a) Základní deska s modulem PXA270M DIMM Max.
b) Spodní strana základní desky s PCMCIA patičí a zasunutou CF kartou.

2.3 Operační systém OPIE Linux

Zkratka OPIE znamená Open Palmtop Integrated Environment a jde o jednu z vývojových větví prostředí Qtopia od Trolltechu. Jde o Open Source grafický systém, který je určen zejména pro PDA a další malá zařízení. Je spuštěn na kernellu Linux, v našem případě verze 2.6.27. Na rozdíl od OS Linux určeného pro osobní počítače, je prostředí zjednodušeno na maximum, chybí zde například manuálové stránky příkazů (příkaz man). OPIE podporuje některé standardní technologie, jako například XML, Bluetooth, IrDA nebo OBEX (přenos specifických datových formátů, například kontaktních vizitek). Obsahem je řada užitečných aplikací, ať už pro správu adresáře, editaci textu, přehrávání hudby a videí, ftp klient, nebo webový prohlížeč Konqueror. V rámci GNU licence je poskytován zdarma. OPIE Linux se standardně instaluje do nejrůznějších typů smartphonů, PDA zařízení a v našem případě i do vývojového kitu VOIPAC PXA270M [21].

3 PROGRAMOVACÍ JAZYKY PRO OS OPIE

3.1 Jazyk C

Programovací jazyk C se začal vyvíjet v Bellových laboratořích AT&T začátkem 70. let. Standardizován byl v roce 1989 jako ANSI C. Postupem času nahrazoval programovací jazyk Basic, ve kterém byl psán kód pro mikropočítače. V 80. letech byl přijat jako hlavní programovací nástroj pro IBM PC. Postupem času se stal (včetně jeho rozšíření C++) hlavním programovacím jazykem pro aplikace v operačních systémech Unix, Linux a Windows.

Jeho hlavní vlastnosti jsou nízkourovňovost, snadná kompilovatelnost a dostatečná jednoduchost, která z něj dělá mocný nástroj pro všestranné použití. Jeho kód je mnohem čitelnější než například jazyk symbolických instrukcí. Vyniká také snadnou přenositelností na různé druhy architektur. Díky jeho jednoduchosti je překlad rychlý, stačí pouze pár strojových instrukcí na každý příkaz.

Velkou výhodou jazyka C je používání, vytváření a editace knihoven, v nichž jsou zapsány funkce, které se pak používají v daném programu. V praxi to vypadá tak, že v hlavičkových souborech jsou definovány funkce. Tyto soubory se pomocí příkazu *#include* „vloží“ do samotného souboru s hlavním kódem. Většina z těchto hlavičkových souborů je volně stáhnutelná. Taktéž podporuje tvůrčí myšlení a osobitý styl programování. K vyřešení daného úkolu se lze dobrat více cestami [5].

3.2 Jazyk Java

Poprvé byl programovací jazyk Java představen firmou Sun Microsystems v roce 1995. Vychází z jazyka C++, který umožňuje objektově orientované programování. Narozdíl od něj, Java již neumožňuje napsání programu, který by nebyl objektově orientovaný. Oproti C++ je Java také více zjednodušená, i když syntaxe je velmi podobná. Největší výhodou oproti C/C++ je automatické přidělování a uvolňování paměti pomocí tzv. garbage collectoru, absence pointerů a jejich nahrazení referencemi a mnohé další. Z těchto důvodů je Java vhodná pro začátečníky, kteří se chtějí naučit programovat. Na více jak polovině českých vysokých škol, kde se programováním zabývají, se Java učí jako první jazyk, viz tab. 3.1.

Tab. 3.1: Přehled programovacích jazyků vyučovaných na českých VŠ.

Fakulta	Předmět	První vyučovaný programovací jazyk
FEKT VUT v Brně	Počítače a programování 1	Java
FEL ČVUT	Programování / Algoritmizace	Java / C na jednom oboru
FEL ZČU	Základy programování pro elektrotechniku	C
FEI VŠB-TUO	Úvod do programování	Java
FIT VUT v Brně	Základy programování	C
FIT ČVUT	Programování a algoritmizace 1	C
FI MUNI	Programování a algoritmizace	Python
FP VUT v Brně	Algoritmizace a programovací techniky	Pascal
EKF VŠB-TUO	Úvod do programování	Java
FIS VŠE	Základy programování	Java

Pozn. Tabulka byla zpracována na základě dostupných studijních plánů jednotlivých fakult, zveřejněných na jejich webových stránkách.

Programovací jazyk Java se také vyznačuje nezávislostí na určeném hardwaru, neboli je multiplatformní. Samotný překlad je realizován nejprve pomocí převedení do tzv. bytecodu, který je pak zpracován a interpretován speciálním programem Java Virtual Machine (VM). Díky VM je zajištěna přenositelnost mezi jednotlivými operačními systémy a platformami, jelikož se pro každou nastaví VM individuálně. Pro běh programů v jazyku Java je potřeba mít nainstalováno prostředí Java Runtime Environment (JRE), které mimo jiné zahrnuje knihovny a interpret Javy.

Java není jen samotný programovací jazyk, ale díky VM i několik platform, určených jak pro standardní desktopové aplikace (J2SE), tak i pro mobilní zařízení (J2ME), případně pro čipové karty (Java Card). Další početnou rodinu tvoří webové aplikace napsané pomocí Java Script [3, 23].

3.3 Jazyk PHP

Jazyk PHP je představitelem skriptovacích jazyků a je určen především pro programování dynamických webových stránek, které potřebují spojení se serverem. V dnešní době se ovšem používá i pro psaní desktopových aplikací, avšak ne v takové míře, jako jazyky C/C++ a Java. PHP znamená Personal Home Page a jde o výtvar Rasmuse Lerdorfa, který vytvořil v jazyku PERL jednoduchý program, který počítal návštěvnost jeho webu. Z důvodu zátěže serveru byl program přepsán v jazyce C. Po spojení s nástrojem Form Interpreter vzniklo PHP/FI 2.0, které se začalo šířit mezi internetovou komunitu. Masové oblíbenosti se ovšem dočkalo až PHP 3.0, které bylo doplněno o mnoho užitečných funkcí (databáze, objekty, cookies). V současné době je k dispozici verze PHP 5.3.3.

Základem celé filozofie PHP je vykonání daného skriptu na straně serveru tak, aby uživatelská úloha spočívala pouze v zadání a přijetí požadované operace, nejčastěji ve formě HTML kódu. Tím se požadavky na uživatele software a hardware stlačují na minimum. Problém nastává tehdy, pokud je server vyčerpán a nestíhá vyřizovat přijaté požadavky. PHP se hodí pro práci s databázemi, formuláři, případně pro manipulaci se soubory [2, 33].

3.4 Volba vhodného programovacího jazyka pro daný úkol

V předchozích odstavcích byly představeny tři nejpoužívanější programovací jazyky. Operační systém OPIE podporuje všechny tři. Z hlediska povahy zadaného problému, kdy program pro měření kvality hovoru bude spuštěn na stejném zařízení jako PBX Asterisk, není potřeba využívat strukturu klient-server, tudíž se jazyk PHP jeví jako nevhodný.

Jazyk Java i jazyk C jsou oba nezávislé na prostředí a platformě. Pro překlad v jazyce je třeba vhodně nastavit překladač GCC tak, aby překládal na architekturu ARM. U jazyka Java je možné stáhnout JRE přímo pro OS OPIE a architekturu ARM a využívat tak platformu J2ME, případně J2SE for Embedded Survey. Jako vhodnější z těchto dvou jazyků se jeví jazyk C, jelikož povaha daného problému není objektivně zaměřena. Grafické prostředí aplikace lze vytvořit i pomocí jazyka C, například pomocí knihovny ncurses [31]. Jazyk C je taktéž méně náročný na hardware, je tedy výhodnější pro vývojový kit, který není tak výkonný jako desktopové PC.

4. POBOČKOVÁ ÚSTŘEDNA ASTERISK

4.1 Základní charakteristika

Asterisk je open-source řešení pro softwarovou pobočkovou ústřednu (softswitch), které podporuje několik komunikačních protokolů, například SIP, VOIP, H.323 aj. Jde tedy o vhodný základ pro menší až střední VOIP síť, její velikost je prakticky dána použitým hardwarem. Kromě samotné funkce PBX se Asterisk dá použít i jako Gateway (přepínání mezi různými protokoly), interaktivní hlasový průvodce (IVR), automatická obsluha (ACD), voicemail, poskytuje možnost šifrování hovorů, podporuje konference a mnoho dalších. Díky podpoře periferních karet (TDM, ISDN aj.) se dají do počítače připojit i klasické telefony, takže uživatel není odkázán pouze na softwarové klienty.

Domovským operačním systémem Asterisku je Linux, lze jej však zprovoznit i na dalších systémech postavených na unixovém jádře, například BSD, Solaris, OS X aj. Díky emulátoru lze Asterisk spustit i na Windows.

V současné době (květen 2011) je nejnovější stabilní uvolněnou verzí Asterisku 1.8.4, vyvíjí se verze 1.8 beta 5. Ta již podporuje protokol IPv6, komunikační protokol Jabber, dokáže integrovat kalendář a mnohem více.

Problematika PBX Asterisk je poměrně rozsáhlá, vzhledem k jeho velké modulárnosti. Následovat bude popis pouze částí, které byly nezbytné pro splnění zadaného úkolu. Pro jeho realizaci byla použita verze 1.6.2.14. Více o Asterisku lze nalézt v literatuře [1, 35].

4.2 Použité součásti PBX Asterisk

4.2.1 Konfigurační soubory

Jde o soubory, v kterých se definují všechny důležité parametry ústředny. Standardní cesta jejich umístění je v */etc/asterisk*. Pro naše potřeby jsou důležité *cdr.conf*, *cdr_custom.conf*, *extensions.conf* a *sip.conf*. V následujících řádcích budou uvedeny pouze obecné charakteristiky, konkrétní nastavení je popsáno v kapitole 5.

Cdr.conf

Tento soubor slouží pro zapnutí jednotlivých voleb v CDR (Call Detail Records), což je záznam jednotlivých vlastností hovoru, například délky hovoru, čísla kanálu, aj. CDR se používá nejčastěji k zaznamenávání tariface, ale díky poli *userfield*, kde se dají definovat uživatelská data výstupu, ho lze použít k mnoha dalším aplikacím. CDR lze uložit buď v textovém formátu .csv, kde jsou jednotlivé položky odděleny ””, nebo ve formátu MySQL [36].

Cdr_custom.conf

V tomto konfiguračním souboru lze nastavit, které položky budou na výstupu souboru .csv a jak se tento soubor bude jmenovat. Soubor je pak implicitně uložen v adresáři */var/log/asterisk/cdr-custom* [37]. Změna tohoto adresáře je možná úpravou konfiguračního souboru *asterisk.conf*, konkrétně oddílu *[directories]* a řádku *astlogdir => /var/log/asterisk*.

Extensions.conf

Zde se nachází samotný číslovací plán, kde je definováno, co bude vykonáno, pokud se volá na jednotlivé klapky v rámci jednotlivých kontextů. Definují se zde také volání mezi jednotlivými kontexty a volání do veřejné sítě pomocí trunků [39].

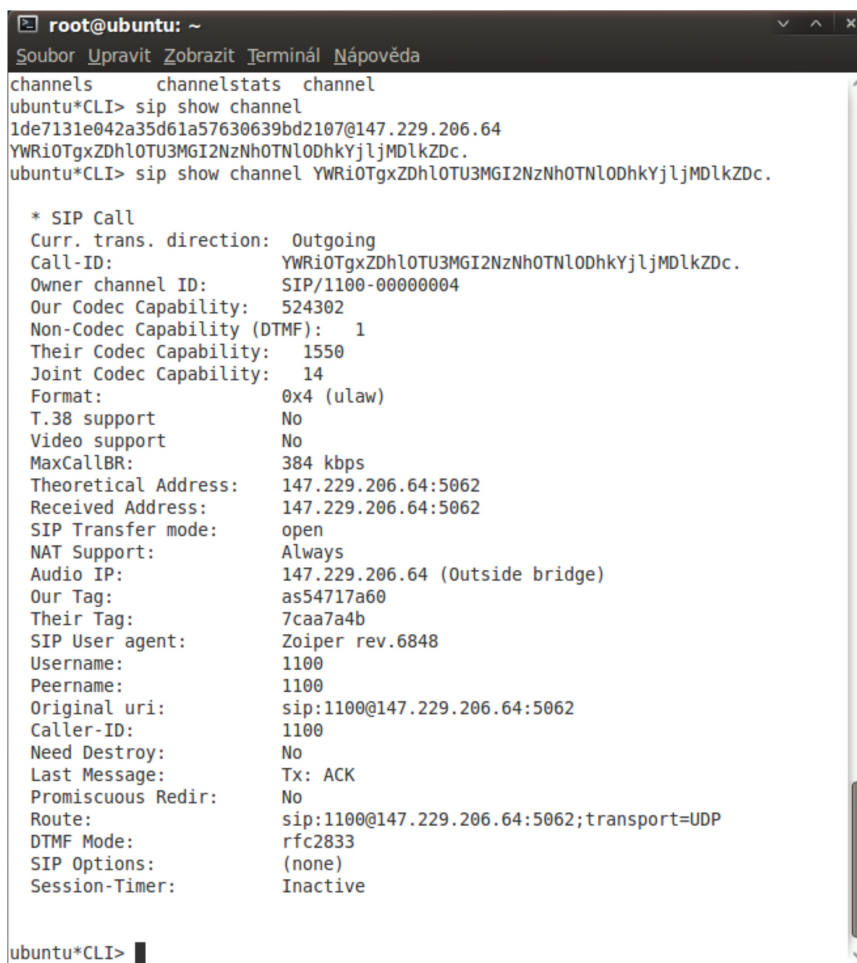
Sip.conf

Konfigurační soubor sloužící k definování SIP účtů, jejich vlastností (například jméno, heslo, typ účtu nebo použitý kodek) a přiřazení k jednotlivým kontextům [40].

4.2.2 CLI

Zkratka CLI znamená Command Line Interface a jedná se defakto o příkazovou řádku PBX Asterisk. V terminálu Linuxu se spouští pomocí příkazu *asterisk -rvvvvvv*, kde počet *v* značí úroveň a podle ní se určuje, kolik informací se v řádce bude zobrazovat. Čím více *v*, tím je výpis podrobnější. Některé funkce, například *NoOp()* zobrazují informace až od určité úrovně, v tomto případě jde o úroveň 3. Všechny příkazy (počet se liší verze od verze), které se dají v CLI použít lze vyvolat pomocí příkazu *help*. V CLI je možné sledovat například přihlašování jednotlivých klientů, námi definované příkazy v *extensions.conf* (například přiřazení hodnot do proměnných), právě vykonávané příkazy a mnoho dalších [38].

Pomocí několika příkazů lze přímo zde sledovat některé informace o kvalitě hovoru, např. pomocí příkazu *rtcp set stats on* nebo *sip show channel*, jak můžeme vidět na obr. 4.1.



```
root@ubuntu: ~
Soubor Upravit Zobrazit Terminál Nápověda
channels channelstats channel
ubuntu*CLI> sip show channel
1de7131e042a35d61a57630639bd2107@147.229.206.64
YWRiOTgxZDhlOTU3MGI2NzNhOTNlODhkYjJlMDlkZDc.
ubuntu*CLI> sip show channel YWRiOTgxZDhlOTU3MGI2NzNhOTNlODhkYjJlMDlkZDc.

* SIP Call
Curr. trans. direction: Outgoing
Call-ID: YWRiOTgxZDhlOTU3MGI2NzNhOTNlODhkYjJlMDlkZDc.
Owner channel ID: SIP/1100-00000004
Our Codec Capability: 524302
Non-Codec Capability (DTMF): 1
Their Codec Capability: 1550
Joint Codec Capability: 14
Format: 0x4 (ulaw)
T.38 support No
Video support No
MaxCallBR: 384 kbps
Theoretical Address: 147.229.206.64:5062
Received Address: 147.229.206.64:5062
SIP Transfer mode: open
NAT Support: Always
Audio IP: 147.229.206.64 (Outside bridge)
Our Tag: as54717a60
Their Tag: 7caa7a4b
SIP User agent: Zoiper rev.6848
Username: 1100
Peername: 1100
Original uri: sip:1100@147.229.206.64:5062
Caller-ID: 1100
Need Destroy: No
Last Message: Tx: ACK
Promiscuous Redir: No
Route: sip:1100@147.229.206.64:5062;transport=UDP
DTMF Mode: rfc2833
SIP Options: (none)
Session-Timer: Inactive

ubuntu*CLI>
```

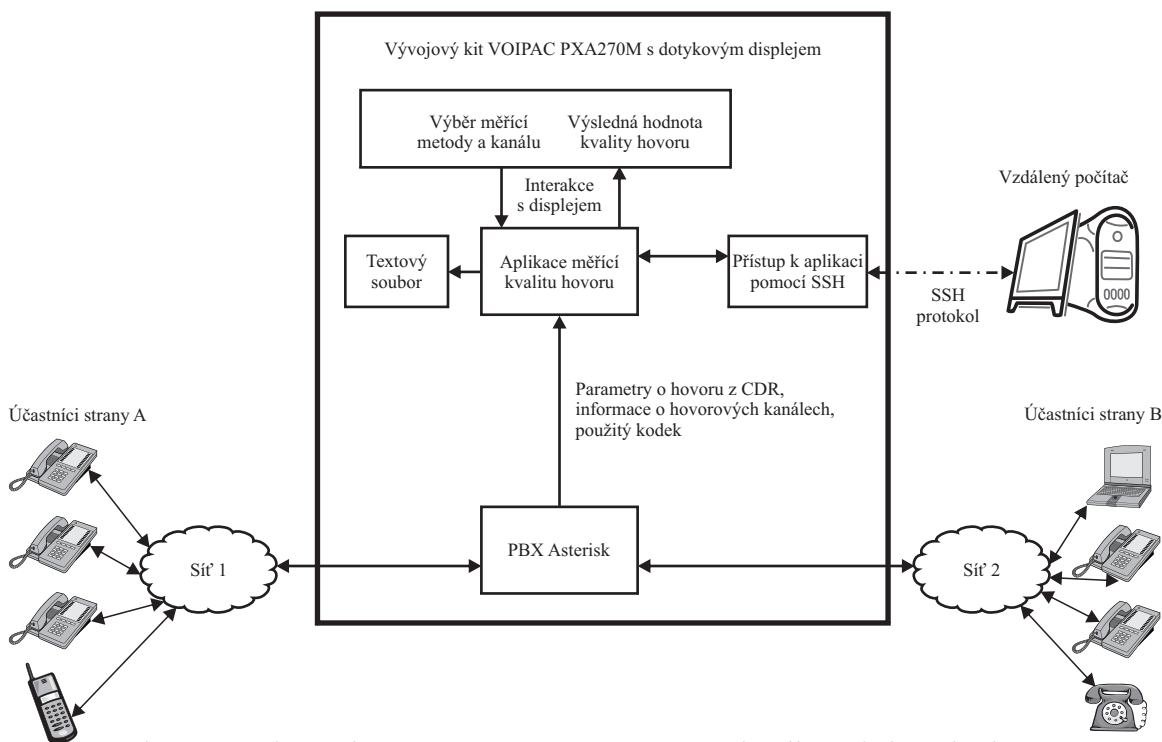
Obr. 4.1: Příkaz sip show channel v CLI.

5 ŘEŠENÍ DANÉHO PROBLÉMU

5.1 Koncepte systému pro měření kvality telefonního hovoru

Koncepce celého měřicího systému v sobě zahrnuje několik základních prvků. Předně je to samotný vývojový kit VOIPAC PXA270M, na kterém je spuštěn PBX Asterisk a aplikace měřící kvalitu telefonního hovoru. Dalšími součástmi jsou účastnická koncová zařízení, ať již hardwarové telefony – klasické analogové, ISDN případně VOIP telefony, nebo bezdrátové telefony založené například na technologii DECT.

Účastnická zařízení mohou být oddělena jednou, či několika sítěmi, PBX Asterisk umožňuje také spojení účastníků, kteří jsou umístěni za routerem. Při každém hovoru jsou jeho parametry zaznamenány pomocí Call Data Record do logového souboru. Z tohoto souboru jsou poté načteny aplikací. Spolu s těmito daty jsou ještě uloženy čísla a kanály obou hovořících stran. Schéma řešení koncepce lze nalézt na obr. 5.1.



Obr. 5.1: Schéma koncepce systému pro měření kvality telefonního hovoru.

Samotná aplikace je založena na prostředí knihovny ncurses, které je uživatelsky přehledné, hardwarově nenáročné a tudíž vhodné pro embedded systémy. Aplikace nejdříve načte příslušná data o hovoru a hovorových kanálech a také identifikaci jednotlivých účastníků hovoru. Uživatel vybere pomocí jednoduchého ovládání hovor, který má být změřen pomocí daného algoritmu. Podle této volby jsou vybrána odpovídající data a spočítán výsledek. Jeho hodnota je zobrazena uživateli pomocí uživatelského prostředí aplikace. Taktéž výsledky vypíše do výstupního souboru. Aplikace je schopna vyhodnocovat několik na sobě nezávislých kanálů pomocí algoritmu E-modelu, kde je R-faktor přepočítáván na hodnotu MOS.

5.2 Nastavení PBX Asterisk

Jak bylo uvedeno v kapitole 4, konfigurační soubory Asterisk se nachází obvykle v */etc/asterisk*. Nastavení výše zmíněných bylo pro ověření funkce aplikace pro měření kvality hovoru následující (v *sip.conf* se dají definovat i více než 4 účastníci):

Cdr.conf

```
[csv]
usegmttime=yes           ;záznam času ve formátu GMT
loguniqueid=yes          ;záznamenávání pod unikátním ID
loguserfield=yes        ;povolení uživatelského pole
```

Cdr_custom.conf

```
[mappings]
Master.csv=>${CSV_QUOTE(${CDR(channel)})},${CSV_QUOTE(${CDR(stchannel)})},${CSV_QUOTE(${CDR(userfield)})} ;do souboru
Master.csv budou uloženy kanály obou účastníků a uživatelské pole
```

Sip.conf

```
[general]
context=kontext1        ;přiřazení všech čísel do kontext1
disallow=all            ;zákaz všech kodeků
allow=alaw              ;povolení kodeku G.711 (příklad)
nat=yes                ;povolení spojení ze sítí za natem

[1100]
type=friend             ;typ účtu pro autentizaci
secret=asdf            ;heslo
username=1100          ;uživatelské jméno
bindport=5060          ;port, na kterém asterisk poslouchá
host=dynamic           ;dynamicky získaná IP adresa
auth=md5               ;metoda ověření
callerid="1100" <1100> ;zobrazení volajícího čísla

[1200]
type=friend             ;typ účtu pro autentizaci
secret=asdf            ;heslo
username=1200          ;uživatelské jméno
bindport=5060          ;port, na kterém asterisk poslouchá
host=dynamic           ;dynamicky získaná IP adresa
auth=md5               ;metoda ověření
callerid="1200" <1200> ;zobrazení volajícího čísla

[1300]
type=friend             ;typ účtu pro autentizaci
secret=asdf            ;heslo
username=1300          ;uživatelské jméno
bindport=5060          ;port, na kterém asterisk poslouchá
host=dynamic           ;dynamicky získaná IP adresa
auth=md5               ;metoda ověření
callerid="1300" <1300> ;zobrazení volajícího čísla
```

```
[1400]
type=friend ;typ účtu pro autentizaci
secret=asdf ;heslo
username=1400 ;uživatelské jméno
bindport=5060 ;port, na kterém asterisk poslouchá
host=dynamic ;dynamicky získaná IP adresa
auth=md5 ;metoda ověření
callerid="1400" <1400> ;zobrazení volajícího čísla
```

Daný kodek je samozřejmě možné měnit. Z Asteriskem podporovaných kodeků však pouze některé mají v doporučení ITU-T G.108 definovanou hodnotu parametru *Ie*. Jejich přehled a hodnota jsou uvedeny v tab. 5.1 a jsou převzaty z [8]. Pokud bude zvolen jiný typ kodeku, aplikace nastaví hodnotu *Iena* nulu.

Tab. 5.1: Hodnoty *Ie* pro podporované hodnoty kodeků v PBX Asterisk.

Typ kodeku	Hodnota <i>Ie</i> [-]
Alaw G.711	0
Ulaw G.711	0
ADPCM G.726 32kb/s	7
CS-ACELP G.729-A + VAD	11
ACELP G.723.1	19
RPE-LTP GSM 06.10 FR	20

V předchozí části bylo uvedeno základní nastavení souborů *cdr.conf*, *cdr_custom.conf* a *sip.conf*. Konfigurační soubor *extensions.conf* a jeho nastavení bude vysvětleno v dalších částech, jelikož při ověřování správné funkce aplikace nebylo jednotné.

Pro správnou funkci Asterisku na vývojovém kitu je ještě třeba změnit cesty pro některé adresáře, které jsou implicitně umístěné ve */var*. Důvodem je implicitní nastavení OS OPIE, kdy je při každém startu systému obsah adresáře */var* přepsán. Tyto cesty se nastaví na začátku souboru *asterisk.conf* následovně:

```
[directories] ; nesmí zde být (!)
astvarlibdir => /xasterisk/lib/asterisk
astdatadir => /xasterisk/lib/asterisk
astlogdir => /xasterisk/log/asterisk
astagidir => /xasterisk/lib/asterisk/agi-bin
astrundir => /xasterisk/run
```

Adresář */xasterisk* je vytvořený v kořenovém adresáři a nachází se zde mj. i aplikace pro měření kvality hovoru.

5.3 Experimentální aplikace pro měření kvality hovoru

Aplikace sloužící k měření kvality hovoru v PBX Asterisk je napsána, dle výsledku rozvahy z kapitoly 3, v programovacím jazyce C. Základní snahou celé aplikace bylo v co největší míře využít možností, které nabízí Asterisk a jeho implementované moduly. Proto byl zvolen postup načítat hodnoty z logového souboru *Master.csv* a zvolený kodek ze souboru *sip.conf*. Tyto dva soubory musí existovat a být umístěny ve svých zdrojových adresářích – */xasterisk/log/asterisk/cdr-custom/* pro *Master.csv* a */etc/asterisk* pro *sip.conf*.

Jako první se inicializují všechny proměnné, kterých bude v dalším průběhu potřeba. Následuje načtení dat z logu Asterisku *Master.csv* a zvoleného kodeku, kde se nejdříve testuje, zda daný soubor vůbec existuje. Pokud tomu tak není, je zobrazeno chybové hlášení a aplikace je ukončena.

Dalším krokem je zobrazení všech dostupných záznamů o hovorech v uživatelském prostředí. Uživatel vidí pořadové číslo hovoru a čísla volajícího a volaného. Jejich počet je omezen použitou knihovnou *ncurses* a velikostí okna na 10. V souboru *Master.csv* tedy nemůže být více jak 10 záznamů o hovoru. Pokud tomu tak je, aplikace další v pořadí již nenačítá. Uživatel poté zvolí jeden z hovorů (graficky se zvýrazní) a svou volbu potvrdí stiskem klávesy šipka vpravo. Pokud chce aplikaci ukončit, zmáčkne klávesu šipka dolů. Směrové klávesy pro potvrzování volby a ukončení aplikace byly zvoleny s ohledem na předem definované proměnné v knihovně *ncurses* a možnosti vývojového kitu, jehož klávesnice není shodná v klávesnici u tradičního PC. Screenshot volby hovoru lze nalézt na obr. 5.2.

Dojde k výběru daného hovoru a do příslušných proměnných se uloží hodnoty zpoždění nutné pro výpočet. Jde o hodnotu *rtt*, která odpovídá veličině *Trv* [7]. Z ní se dále vypočítají hodnoty *T* a *Ta*. Zbývající hodnoty nutné pro výpočet jsou převzaty z telekomunikačního věstníku TSB-116-A z března 2006 [30]. Jejich hodnoty jsou uvedeny v tab. 5.2.

Následně se vypočítají příslušné hodnoty. Z dílčích hodnot se poté počítá R-faktor, který je dále přepočítán podle vzorce (1.29) na MOS. Jsou zobrazeny výsledky a hodnota MOS spolu s identifikací hovoru je zapsána do výstupního souboru *MOS.txt*. Pro návrat k výběru hovoru slouží stisk klávesy šipka vlevo. Ukončení aplikace se provede pomocí klávesy šipka dolů. Screenshot výsledného výpočtu je vidět na obr. 5.3. Zjednodušený vývojový diagram celé aplikace lze nalézt v příloze 2, úplný vývojový diagram v souboru *vyvojovy_diagram.pdf* na přiloženém CD. Zdrojový kód je dostupný na přiloženém CD.

Tab. 5.2: Hodnoty parametrů sloužících k výpočtu dle literatury [35].

Název	Jednotka	Hodnota
SLR _S	dB	8
RLR _R	dB	2
STM _R	dB	15
LSTR	dB	18*
D _s	-	3
D _r	-	3
TEL _R	dB	65
WEPL	dB	110
q _{du}	-	1
B _{pl}	-	1
P _{pl}	%	0
N _c	dBm0p	-70
N _{for}	dBmp	-64
P _s	dB(A)	35
P _r	dB(A)	35
*LSTR = STM _R +D _r		

```

Experimentalni aplikace pro vypocet MOS

Seznam nactenych hovorů

Volajici  Volany
Hovor cislo 1  1100  1200
Hovor cislo 2  2400  1000
Hovor cislo 3  1300  1400
Hovor cislo 4  1200  1400

Zvol cislo hovoru a pote stiskni sipku vpravo
(c)2010 Bc. Petr Bilek          Pro ukončení programu, zmackni sipku dolu

```

Obr. 5.2: Volba hovoru, u kterého má být vypočítána kvalita.

```

Experimentalni aplikace pro vypocet MOS

R = R0 - Is - Id - Ie + A
88.250412 = 94.739143 - 4.810214 - 11.298942 - 0 + 0

MOS = 1 + 0,035 * R + R * (R - 60) * (100 - R) * 7E-6
MOS = 4.293816

Pro navrat na vyber hovoru zmackni sipku vlevo
(c)2010 Bc. Petr Bilek          Pro ukončení programu, zmackni sipku dolu

```

Obr. 5.3: Výsledný výpočet kvality hovoru.

5.4 Implementace aplikace do vývojového kitu

Aplikace měřící kvalitu hovoru byla programována, testována a laděna ve vývojovém prostředí Netbeans 6.9.1 v operačním systému Ubuntu Linux 9.10. Testování probíhalo v návaznosti na nainstalovanou verzi Asterisku 1.6.2.2, která byla předinstalována i na vývojovém kitu. Vývojový kit má jinou architekturu než standardní PC. Je postaven na procesoru ARM. Pro tuto architekturu bylo třeba přeložit i celou aplikaci. Nástroje gcc a g++ mají svou obdobu i pro ARM architekturu. Konkrétně byly použity nástroje arm-linux-gcc a arm-linux-g++ z balíku arm-linux-gcc toolchain 3.4.1. Správnost překladač byla ověřena v příkazové řádce pomocí příkazu *file jmeno_overovaneho_souboru*. Ve výpisu byla zobrazena konkrétní architektura ARM.

Po překladu byla aplikace zkopírována na kit do adresáře */xasterisk*. Při spuštění se ovšem objevil problém s knihovnou *ncurses* a terminálem operačního systému OPIE. Terminál nedokáže správně zobrazit uživatelské rozhraní aplikace. Správného zobrazení aplikace lze dosáhnout pomocí vzdáleného přístupu. Tento problém se do termínu odevzdání této práce nepodařilo odstranit.

I přes tento zjevný nedostatek bylo potřeba ověřit správnou funkčnost aplikace. Byla k tomu využita PC verze, které byl dodán log soubor vytvořený Asteriskem, který je nainstalován v kitu. Výsledky výpočtů z následující kapitoly jsou výstupem aplikace v PC verzi, na jejich hodnoty nemá změna architektury vliv.

5.5 Ověření funkčnosti a správného návrhu aplikace

5.5.1 Volání na klapku pomocí softwarového klienta

Jako první a nejjednodušší způsob ověření funkčnosti aplikace, který lze realizovat pomocí notebooku a vývojového kitu, bylo volání softwarovým klientem na danou klapku. Na notebooku byla předinstalována nejnovější verze (2.15) softwarového klienta Zoiper. Nastavení souboru *extensions.conf* bylo následující:

Extensions.conf

```
[kontext1]
exten =>4,1,Answer() ;vzvednutí hovoru
exten =>4,n,Wait(3) ;čekání 5 vteřin
exten =>4,n,Saydigits(111111) ;přehraje text
exten =>4,n,Set(a=${CHANNEL(rtpqos,audio,all)})
;nastavení proměnné a
exten =>4,n,Set(CDR(userfield)=${a})
;do userfield obsah proměnné a
exten =>4,n,ResetCDR(vw) ;reset CDR
exten =>4,n,NoCDR() ;zabraňuje zápisu do CDR
exten =>4,n,HangUp() ;zavěšení
```

Výsledek logu v souboru *Master.csv* pak vypadá takto:

```
"SIP/1100-00000002","", "ssrc=2080375489;themssrc=4126164670;lp=0;rxjitter=0.000000
0;rxcount=1;txjitter=0.000000;txcount=0;rlp=0;rtt=0.000000"
```

Je vidět, že dle nastavení v *cdr_custom.conf* se zde nachází kanál volaného, společně s jeho číslem (první 4 čísla za SIP/ jsou číslo, čísla za pomlčkou pak identifikace kanálu) a obsah pole *userfield*, ve kterém se nachází obsah funkce *Channel*. Význam je následující:

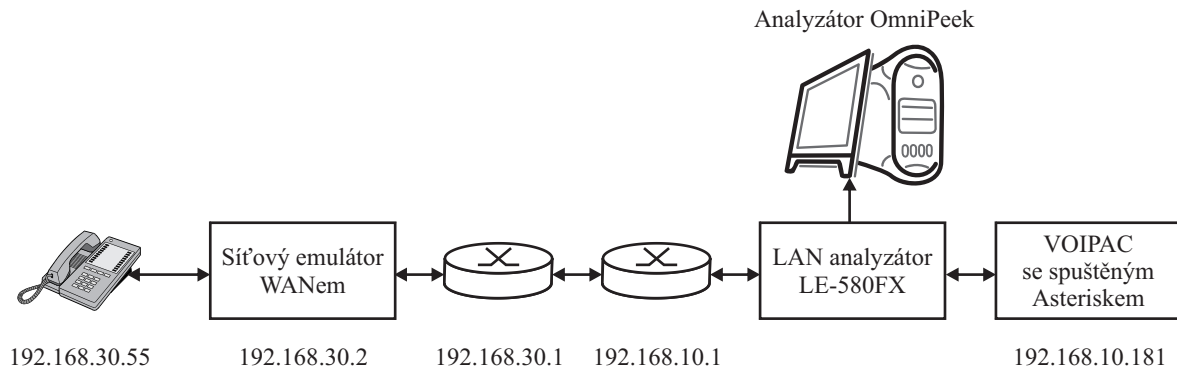
- *ssrc* – lokální synchronizační číslo,
- *themssrc* – vzdálené synchronizační číslo,
- *lp* – ztracené pakety na lokální straně,
- *rxjitter* – jitter pro přijímané pakety,
- *rxcount* – počet přijatých paketů,
- *txjitter* – jitter pro vysílané pakety,
- *txcount* – počet vyslaných paketů,

- rlp – počet paketů ztracených na vzdálené straně,
- rtt – čas smyčky, kterou signál urazí od vysílací strany k přijímací a zpět.

Nulová hodnota pole *rtt* zapříčiní ve výsledném výpočtu chybu, jelikož se defakto dosazují doporučené hodnoty (naleznutelné v [8, 30]) a výsledná hodnota MOS se blíží své horní hranici. Chyba je zapříčiněna jednak absencí telekomunikačního řetězce, notebook a vývojový kit byly spojeny křížovým kabelem „napřímo“ a jednak použitím softwarového klienta, který nevysílá potřebná data, která Asterisk zpracovává při výpočtu *rtt*. Podařilo se však ověřit, že funguje zápis do logového souboru.

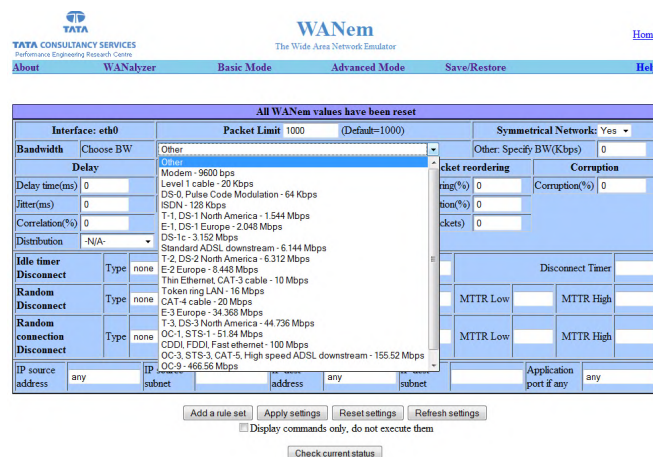
5.5.2 Volání na klapku pomocí VoIP telefonu Well 3130F

Jelikož volání v předchozím zapojení nepřineslo uspokojivé výsledky, bylo nutné pro ověření celé zapojení přepracovat a doplnit o telekomunikační síť. Pro simulaci různého zatížení sítě posloužil síťový emulátor WANem [28]. Pro kontrolu správných výsledků měření byl použit LAN analyzátor Lineeye LE-580FX [15] a softwarový analyzátor OmniPeek [20], který dokáže analyzovat RTP stream a vypočítat z něj hodnoty R-faktoru a MOS. Zapojení celé měřicí struktury je vidět na obr. 5.4.



Obr. 5.4: Měřicí struktura při volání VoIP telefonem na klapku.

Měření probíhalo následovně – po zapojení celého pracoviště a správné konfiguraci všech jeho prvků se na vývojovém kitu spustil PBX Asterisk. Číslovací plán v souboru *extensions.conf* zůstává stejný, jako v kapitole 5.4.2. Na webové adrese <http://192.168.30.2/WANem> se nalézá uživatelské prostředí emulátoru. Zde se dají nastavovat různé parametry emulované sítě, přes které bude uskutečněn hovor (viz obr. 5.5).

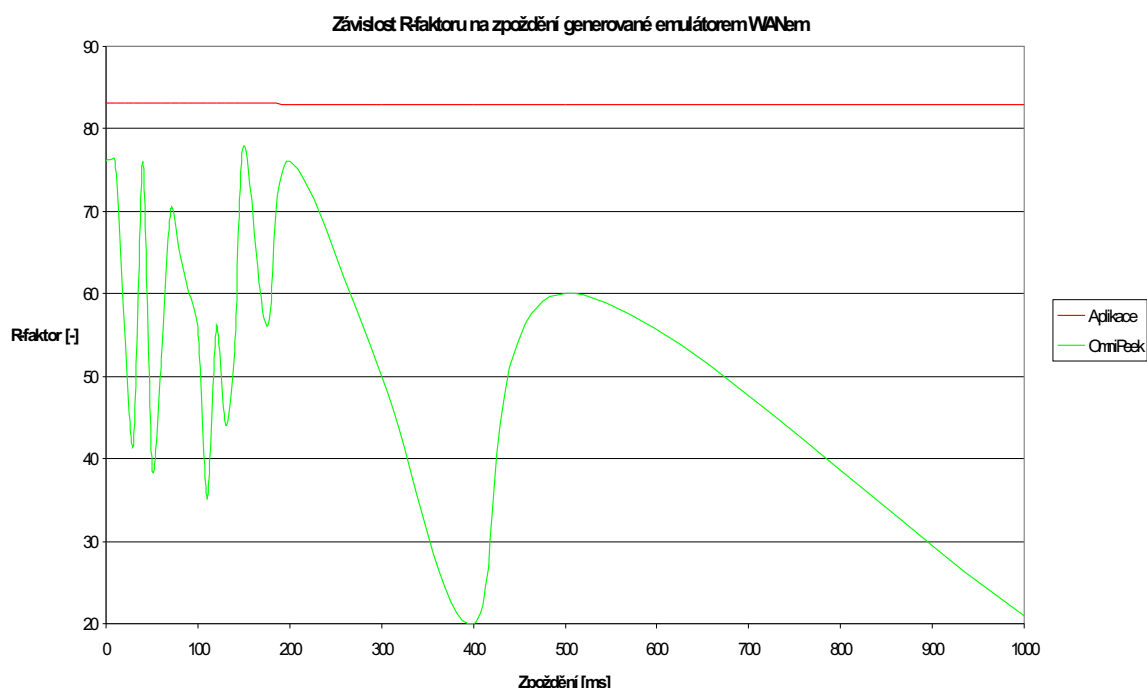


Obr. 5.5: Uživatelské prostředí emulátoru WANem.

Voláním na danou klapku se spustí předem definovaný zvuk, který je přehráán. Do log souboru jsou zaznamenány hodnoty volajícího čísla a pole userfield. Asterisk dokáže do log souboru zaznamenávat jak zpoždění, tak i jitter. Ve vzorcích v doporučení ITU-T G.107 [7] se ovšem uvažuje pouze o vlivu zpoždění (viz kapitola 1.7). V emulátoru WANem byla tedy postupně zvyšována hodnota zpoždění a zaznamenávány výsledky ze souboru *Master.csv*, které poté byly předány aplikaci. S její pomocí byla vypočítána hodnota MOS. Tyto hodnoty byly poté srovnány s hodnotou MOS-CQ, kterou vypočítal analyzátor OmniPeek. Emulovaná síť byla nastavena na 100Mb Fast Ethernet, použit byl kodek G.711 alaw. Naměřené hodnoty lze nalézt v tab. 5.3. Grafickou závislost pak na obr. 5.6 a 5.7.

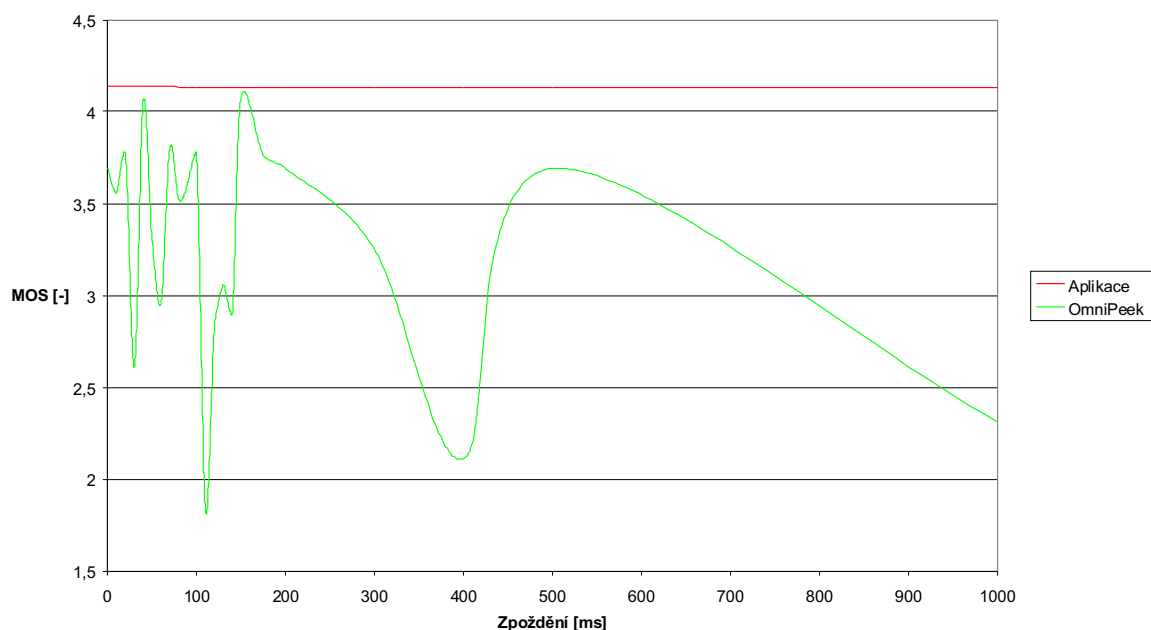
Tab. 5.3: Hodnoty rtt a MOS naměřené při volání na klapku.

pro alaw, SIP, 100Mbps Ethernet, volání na klapku 4, Saydigits (11111)					
Delay WANem [ms]	rtt aplikace [ms]	R-faktor aplikace [-]	MOS aplikace [-]	R G.107 OmniPeek [-]	MOS-CQ OmniPeek [-]
0	23,492	83,123024	4,136375	76	3,69
10	23,799	83,092453	4,135332	76	3,56
20	23,834	83,088936	4,135212	56	3,76
30	23,873	83,085014	4,135078	42	2,61
40	23,910	83,081284	4,134951	76	4,06
50	23,945	83,077744	4,13483	39	3,3
60	23,987	83,073494	4,134685	52	2,96
70	24,022	83,069939	4,134563	70	3,8
80	24,055	83,066589	4,134449	65	3,52
90	24,089	83,063126	4,134331	60	3,61
100	24,114	83,060577	4,134243	56	3,76
110	24,147	83,057213	4,134129	35	1,83
120	24,172	83,054657	4,134041	56	2,81
130	24,203	83,051476	4,133933	44	3,06
140	24,234	83,048302	4,133824	53	2,91
150	24,275	83,044090	4,13368	78	4,08
175	24,317	83,039764	4,133532	56	3,76
200	24,345	83,036873	4,133433	76	3,69
300	24,380	83,033257	4,133309	50	3,25
400	24,407	83,030464	4,133214	20	2,11
500	24,438	83,027252	4,133104	60	3,69
1000	24,503	83,020493	4,132873	21	2,31



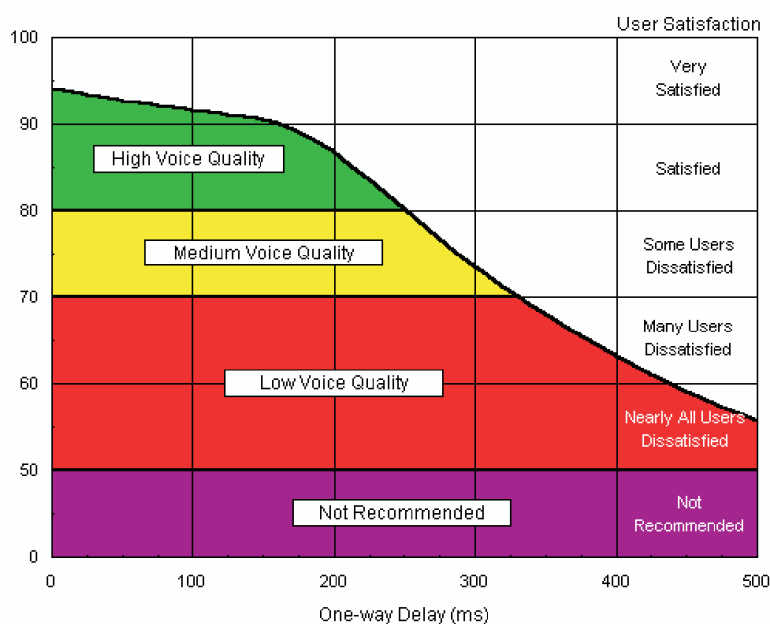
Obr. 5.6: Naměřená závislost R-faktoru na zpoždění při volání na klapku.

Závislost MOS na zpoždění generované emulátorem WANem



Obr. 5.7: Naměřená závislost MOS na zpoždění při volání na klapku.

Z tab. 5.3 a grafických závislostí na obr. 5.6 a 5.7 je patrné, že se vzrůstajícím zpožděním se hodnota *rtt*, kterou generuje Asterisk do log souboru, prakticky nemění (roste jen velmi nepatrně). Tím pádem je R-faktor prakticky po celou dobu roven hodnotě 83 a MOS 4,1. Tato hodnota odpovídá velmi dobré kvalitě signálu. Již při měření byl tento údaj označen za chybný, jelikož při zpoždění přesahujícím 400 ms byla subjektivně patrná degradace signálu a jeho zpoždění v telekomunikačním řetězci. Srovnání s hodnotami R-faktoru a MOS, které vypočítal analyzátor OmniPeek nelze považovat za průkazné. Tyto hodnoty jsou velmi rozkolísané a taktéž neodpovídají teoretickým předpokladům. Referenční grafická závislost, která je převzatá z literatury [30], je k vidění na obr. 5.8.



Obr. 5.8: Referenční závislost R-faktoru na zpoždění signálu.

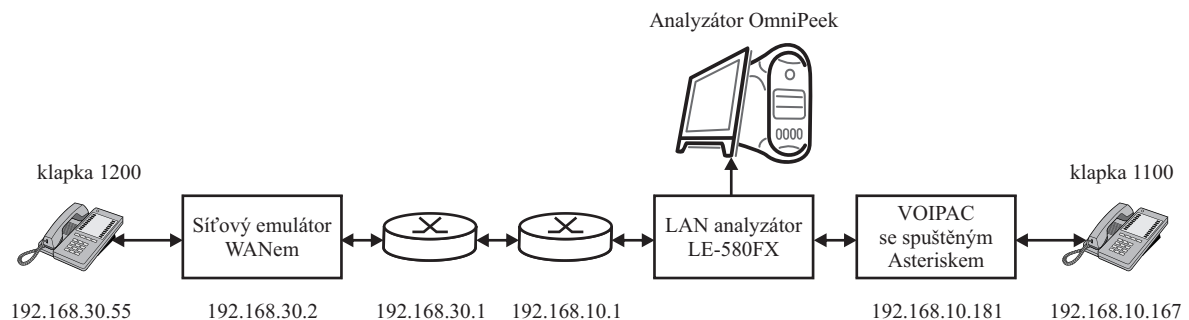
5.5.3 Volání namezi dvěma hardwarovými telefony Well

Při druhém experimentálním ověření správnosti návrhu aplikace byla již sestavena celá hovorová cesta, tzn. se dvěma účastníky. Ostatní síťové prvky byly ponechány ve stejné konfiguraci jako v předchozím případě. Celý systém lze nalézt na obr. 5.9. Musel se ovšem změnit číslovací plán v souboru *extensions.conf*, kde je definováno volání mezi dvěma účastníky:

```
[kontext1]
exten =>1200,1,Set(b=1200) ;nastavení kontrolní proměnné
exten =>1200,2,Dial(SIP/1200,100000,g) ;volání na kl. 1200
exten =>1200,3,Verbose(${b}) ;zobrazení kontrolní proměnné
;zbytek stejný význam jako v předchozím
exten =>1200,n,Set(a=${CHANNEL(rtpqos, audio, all)})
exten =>1200,n,Set(CDR(userfield)=${a})
exten =>1200,n,ResetCDR(vw)
exten =>1200,n,NoCDR()

exten =>1100,1,Set(b=1100) ;nastavení kontrolní proměnné
exten =>1100,2,Dial(SIP/1100,100000,g) ;volání na kl. 1100
exten =>1100,3,Verbose(${b}) ;zobrazení kontrolní proměnné
;zbytek stejný význam jako v předchozím
exten =>1100,n,Set(a=${CHANNEL(rtpqos, audio, all)})
exten =>1100,n,Set(CDR(userfield)=${a})
exten =>1100,n,ResetCDR(vw)
exten =>1100,n,NoCDR()
```

V příkazu *Dial()* je nutné mít parametr *g*, který nechává dále provádět příkazy následující pod příkazem *Dial()*, pokud daný účastník zavěsil.



Obr. 5.9: Měřicí struktura při volání mezi dvěma VoIP telefony.

Z číslovacího plánu by se mohlo zdát, že nezáleží na tom, v jakém směru hovor povedeme. Obě účastnické klapky mají u svého volání záznam do logového souboru. Je ovšem nutné vzít v úvahu zapojení celé telekomunikační cesty. Pro měření v závislosti na hodnotách zpoždění generovaného emulátorem WANem je nutné volat z klapky 1200 na 1100. V analyzátoru OmniPeek, je pak vybrán RTP stream mezi IP adresou 182.168.10.181 a 192.168.30.55. Důvodem je, že při opačném směru by analyzátor zachytával nezpožděné pakety, které neprošly emulátorem WANem. Celková délka každého testovaného hovoru byla

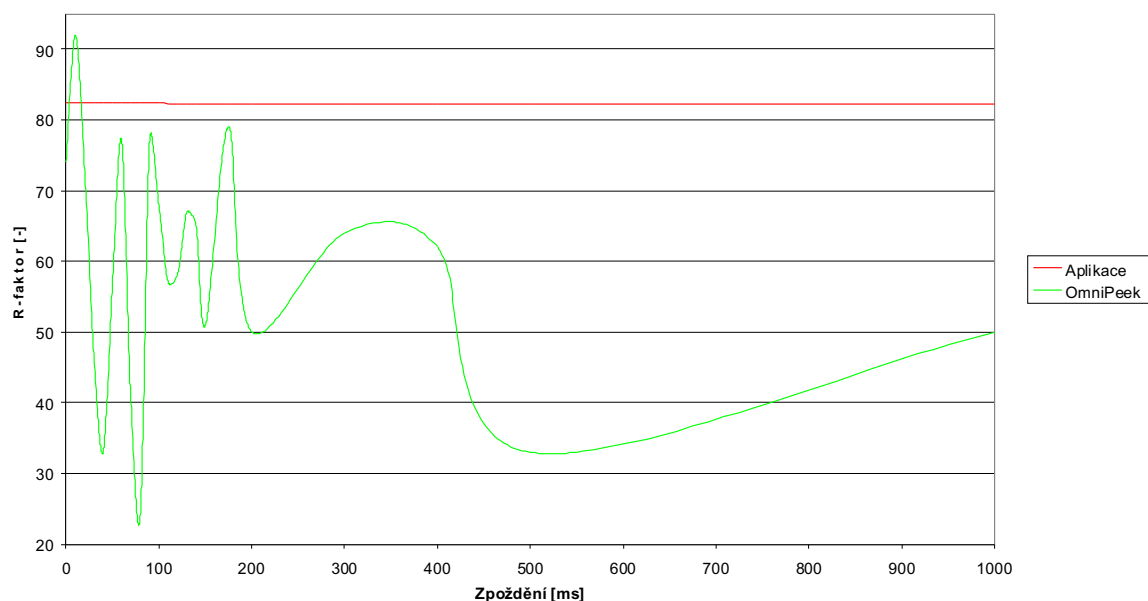
10 ms. Stejně, jako v předchozím případě, byl použit kodek G.711 alaw a v emulátoru WANem byl typ sítě nastaven na 100Mb Fast Ethernet. Naměřené hodnoty lze nalézt v tab. 5.4, grafické závislosti na obr. 5.10 a 5.11.

Tab. 5.4: Hodnoty *rtt* a MOS naměřené při volání mezi dvěma účastníky.

pro alaw, SIP, 100Mbps Ethernet, délka hovoru 10 s					
Delay WANem [ms]	rtt aplikace [ms]	R-faktor aplikace [-]	MOS aplikace [-]	R G.107 OmniPeek [-]	MOS-CQ OmniPeek [-]
0	29,900	82,366684	4,110231	74	4,03
10	29,989	82,354111	4,109791	92	4,17
20	30,046	82,346024	4,109507	73	3,87
30	30,079	82,341331	4,109343	47	3,56
40	30,104	82,337769	4,109218	33	1,92
50	30,131	82,333916	4,109083	57	3,2
60	30,154	82,330627	4,108967	77	4,06
70	30,176	82,327477	4,108857	40	3,3
80	30,208	82,322891	4,108696	24	2,31
90	30,235	82,319016	4,10856	77	3,97
100	30,270	82,31398	4,108383	68	3,97
110	30,294	82,310516	4,108262	57	3,8
120	30,323	82,306335	4,108115	58	3,69
130	30,351	82,302284	4,107973	67	3,76
140	30,380	82,29808	4,107825	65	3,61
150	30,406	82,294312	4,107693	51	3,1
175	30,429	82,29097	4,107575	79	4
200	30,461	82,286308	4,107411	50	3,69
300	30,485	82,282814	4,107289	64	3,84
400	30,517	82,278137	4,107124	62	3,73
500	30,542	82,274483	4,106996	33	2,36
1000	30,573	82,269936	4,106836	50	3,73

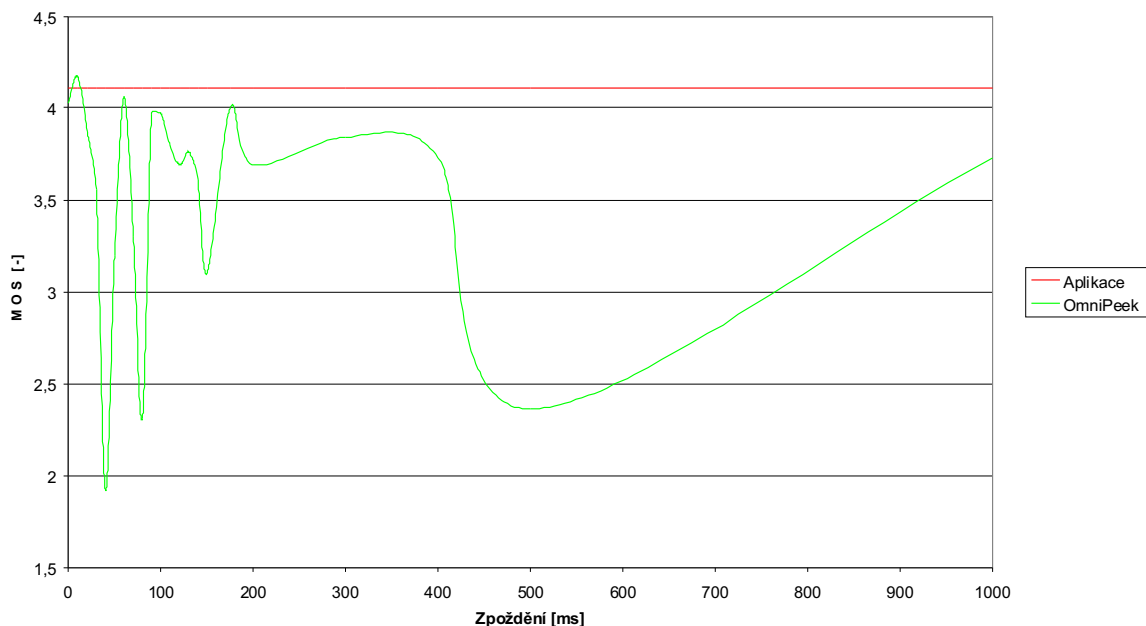
Z naměřených hodnot a grafů pozorujeme stejný průběh obou závislostí, jako při volání na klapku. Velikost *rtt* se prakticky nemění a zůstává kolem hodnoty 30 ms. Díky tomu je hodnota R- faktoru přibližně 82, stejně jako hodnota MOS 4,1. Při pohledu na tab. 5.3 a 5.4 vidíme, že při zvýšení hodnoty *rtt* cca o 6 ms se hodnota R-faktoru a MOS sníží jen o setiny až desetiny. Nelze uspokojivě porovnat výsledky z aplikace s měřením analyzátoru OmniPeek, jelikož jeho výsledné hodnoty jsou velmi rozkolísané. Následuje porovnání výsledku měření s referenčním průběhem na obr. 5.8. Z tohoto porovnání jasně vyplývá, že výsledky tohoto měření neodpovídají teoretickým předpokladům.

Závislost R-faktoru na zpoždění generované emulátorem WANem



Obr. 5.10: Naměřená závislost R-faktoru na zpoždění při volání mezi účastníky.

Závislost MOS na zpoždění generované emulátorem WANem



Obr. 5.11: Naměřená závislost MOS na zpoždění při volání mezi účastníky.

5.6 Zhodnocení současné koncepce

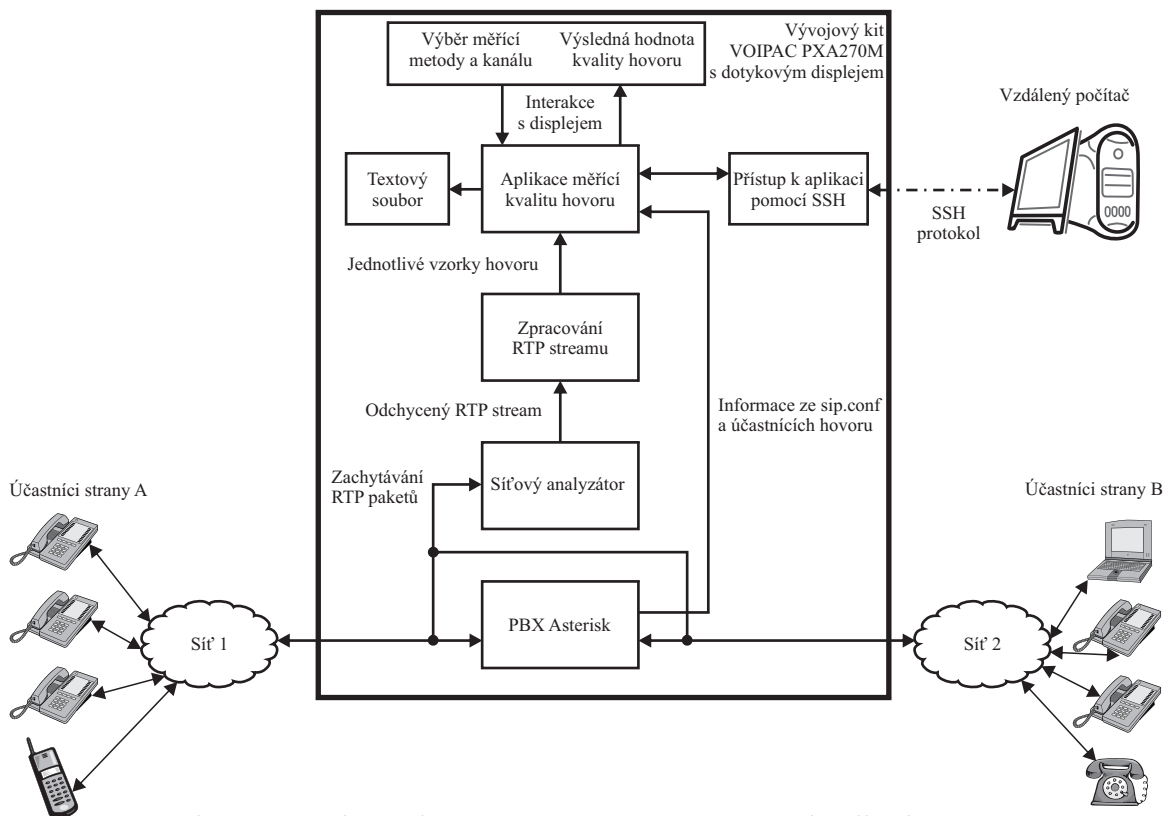
V minulé kapitole byl pomocí síťového emulátoru WANem testován správný návrh celé koncepce, potažmo aplikace měřící kvalitu hovoru. Výsledky se rozcházejí s teoretickým předpokladem a mezinárodními doporučeními. Aplikace využívá jako zdrojová data hodnotu *rtt* ze souboru *Master.csv*, což je log souboru PBX Asterisk. Tato proměnná dle technické dokumentace [17] znamená round trip time. Dle předpokladu by měla odpovídat stejnojmenné veličině v doporučení ITU-T G.107 [7] a měl by se z ní dát vypočítat faktor zhoršení vlivem zpoždění signálu, které podle vzorců v [7] ovlivňuje celkový výpočet. Z výsledků měření, které jsou popsány v minulé kapitole, lze vidět, že hodnota *rtt* neodpovídá předpokládané hodnotě dvojnásobku času potřebného k průchodu signálu jedním směrem v telekomunikační cestě, tudíž je pro výpočet nepoužitelná.

Celá koncepce byla navržena tak, aby, co nejvíce využívala vlastností a dat z PBX Asterisk, kde se zdálo nejvhodnější využít výše zmíněný log soubor. Ten ještě obsahuje hodnoty jitteru a počet ztracených paketů. Tyto hodnoty ovšem nejsou při výpočtu R-faktoru dle [7] využity (jitter), případně jsou již obsaženy v parametru *Ie* (ztrátovost). Lze tedy tvrdit, že koncepce měření kvality hovoru, jež je založená na datech poskytovaných PBX Asterisk v současnosti vykazuje neuspokojivé výsledky.

Možným lepším řešením by bylo přistoupit k řešení celé problematiky nezávisle na PBX Asterisk. Využít síťový analyzátor, který bude odchyťvat pakety na vstupu/výstupu Asterisku, vyhodnocovat je a případně RTP pakety dále zpracovávat. Koncepci tohoto řešení ukazuje obr. 5.12. Aplikace by pak zpracovávala jednotlivé časové značky mezi pakety a z nich počítala zpoždění, z kterého by pak bylo možné počítat hodnotu R-faktoru a MOS.

Při tomto řešení, jehož základem je paketový analyzátor, by bylo možné implementovat další měřicí algoritmy, například 3SQM. Problémem celého řešení je časová náročnost

jeho realizace a problém s implementací do vývojového kitu. Jelikož je třeba využívat „čistého“ jazyka C (například v jazyce Java jsou předem definované knihovny pro paketovou analýzu, ale implementace JRE na VOIPAC se jeví jako složitý a komplexní problém) je výběr knihoven dosti omezený. Existuje knihovna libpcap [29], která je základem například analyzátoru Wireshark. Pro její správnou funkci je ovšem potřeba několika dalších knihoven, například lex a bison. Přeložit všechny tyto knihovny pro vývojový kit je možné, ale velmi náročné, jelikož jsou na sobě závislé a každá by se pro ARM musela kompilovat zvlášť. Koncepte založená na síťovém analyzátoru nebyla již, z časových důvodů, realizována. Experimentálně byla realizována a ověřena funkce libpcap na PC.



Obr. 5.12: Schéma koncepce systému pro měření kvality hovoru.

Případná implementace dalších měřících algoritmů, například 3SQM již není zřejmě tak složitá. Doporučení P.563 [10] obsahuje i volně šiřitelný kód metody, který je ovšem potřeba upravit pro danou koncepci. Největším problémem se ovšem jeví vývoj a implementace paketového analyzátoru do vývojového kitu VOIPAC PXA270M.

6 ZÁVĚR

V této diplomové práci byla rozebrána problematika měření kvality telefonních hovorů. V úvodu teoretické části byly popsány metody, které se pro měření kvality využívají. Nejprve byla popsána metodika měření pomocí stupnice MOS. Následně byly metody rozděleny na subjektivní a objektivní. Subjektivní metody jsou založeny na hodnocení hovoru určitou skupinou osob, které jej popisují na základě svých osobních vjemů. Jsou však časově i finančně nákladné. Poskytují ovšem nejlepší výsledky. Naproti tomu objektivní metody využívají algoritmů, které jsou zpracovány výpočetní technikou a jejichž výstupem je hodnota MOS, která se blíží hodnotám měřeným pomocí subjektivních metod. Poslední možností měření kvality telefonního hovoru je takzvaný E-model, ve kterém se uvažují parametry přenosového telekomunikačního řetězce. Výsledkem je hodnota zvaná R-faktor, která se následně přepočítává na MOS.

Dále byl v teoretické části vysvětlen pojem embedded systém a popsán zástupce těchto zařízení – vývojový kit VOIPAC PXA 270M. Byla popsána jeho architektura, základní hardwarové komponenty a operační systém OPIE Linux.

Poslední kapitolou v teoretické části je popis softwarové pobočkové ústředny Asterisk, která v sobě kombinuje funkce pobočkové ústředny, meziprotokolové brány, interaktivní hlasové odezvy a mnoha dalších. Byly popsány nejdůležitější komponenty a moduly, které se dále používají v praktické části této práce.

Na začátku praktické části je rozebrána problematika programovacích jazyků, které připadaly v úvahu pro implementaci softwaru do vývojového kitu VOIPAC PXA 270M. Bylo rozhodováno mezi jazyky C, Java a PHP. Byla provedena diskuze, ze které vyplynulo jako ideální řešení použít programovací jazyk C. Tento jazyk je nezávislý na platformě, je velmi intuitivní a výsledný program je hardwarově nenáročný, hodí se tedy právě pro implementaci v embedded systémech.

V praktické části je dále popsána autorem vytvořená koncepce systému pro měření kvality telefonního hovoru, jehož základem je vývojový kit VOIPAC PXA 270M a PBX Asterisk. Na základě výše uvedené koncepce byly vytvořeny zdrojové kódy aplikace v jazyce C. Aplikace měřící kvalitu telefonního hovoru využívá záznamový (log) soubor Asterisku, ve kterém jsou uloženy důležité parametry popisující průběh uskutečněného hovoru. Uživatel si zvolí konkrétní hovor a z daných hodnot je podle mezinárodního doporučení ITU-T G.107 aplikací vypočítána hodnota R-faktoru, která je přepočítána na hodnotu MOS. Výsledek je zobrazen uživateli aplikace v grafickém prostředí aplikace. Do výstupního souboru *MOS.txt* je uložena hodnota parametru MOS společně s čísly volajícího a volaného.

Koncepce byla dále experimentálně ověřena pomocí měřicího systému, kde zátěž představoval síťový emulátor WANem a k porovnání s výstupními hodnotami aplikace sloužil síťový analyzátor OmniPeek. Výsledky tohoto ověření ukazují, že PBX Asterisk předává aplikaci špatnou hodnotu proměnné *rtt*. Ta neodpovídá stejnojmenné veličině v doporučení ITU-T G.107, ze které se dále počítá R-faktor. Výsledky experimentálního měření jsou v tomto případě velmi zkrácené a neodpovídají teoretickým předpokladům (viz obr. 5.8). Schéma experimentálních zapojení a výsledky dílčích měření lze nalézt v kapitole 5.5.

Implementace do vývojového kitu neproběhla zcela bez problémů. Ukázalo se, že terminál není schopen správného zobrazení knihovny ncurses, ověření tedy probíhalo z log souboru vytvořeného ve vývojovém kitu, ale pomocí aplikace v PC verzi.

Závěrečná část práce je věnována návrhu úpravy stávající koncepce, kde byla navržena změna, která by mohla vést ke správnému počítání hodnoty R-faktoru a dále by umožnila rozšíření celé aplikace o další měřicí metody. Tato inovovaná koncepce nebyla z časových důvodů, již plně realizována. Experimentálně byla realizována část zachytávající síťový provoz pomocí knihovny libpcap na PC.

LITERATURA

- [1] Asterisk - The Open Source Telephony Projects [online]. 2010 [cit. 2010-12-14]. Asterisk. Dostupné z WWW: <<http://www.asterisk.org/>>.
- [2] BRÁZDA, Jiří. PHP 4 - učebnice základů jazyka. Praha : Grada Publishing a. s., 2002. 224 s. ISBN 80-247-0442-0.
- [3] Dione - studentský informační server ZČU [online]. 2006 [cit. 2010-12-14]. Programovací jazyk Java. Dostupné z WWW: <<http://v1.dione.zcu.cz/java/uvod.html>>.
- [4] HEATH, Steve. Embedded systems design. [s.l.] : Newnes, 2003. 430 s. ISBN 978-07-5065-546-0.
- [5] HEROUT, Pavel. Učebnice jazyka C - 1. díl. České Budějovice : Kopp, 2008. 270 s. ISBN 978-80-7232-351-7.
- [6] IT POINT [online]. 2008 [cit. 2010-12-14]. Měření a hodnocení QoS v IP telefonii. Dostupné z WWW: <<http://www.itpoint.cz/ip-telefonie/teorie/mereni-hodnoceni-qos-ip-telefonie.asp>>.
- [7] ITU-T G.107: The E-Model, a computational model for use in transmission planning, 2003.
- [8] ITU-T G.108: Application of the E-model: A planning guide, 1999.
- [9] ITU-T P.10: Vocabulary of terms on telephone transmission quality and telephone sets, 1998.
- [10] ITU-T P.563: Single-ended method for objective speech quality assessment in narrow-band telephony applications, 2004.
- [11] ITU-T P.800.1: Mean Opinion Score (MOS) terminology, 2003.
- [12] ITU-T P.861: Methods for objective and subjective assessment of quality, 1998.
- [13] ITU-T 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, 2001.
- [14] ITU-T P.862.1: Mapping function for transforming P.862 raw result scores to MOS-LQO, 2003.
- [15] LE-580FX [online]. 2011 [cit. 2011-05-09]. TR Instruments. Dostupné z WWW: <<http://www.trinstruments.cz/le-580fx>>.
- [16] Matlab server FEL [online]. 2004 [cit. 2010-12-14]. Schematické znázornění parametrů pro výpočet R-faktoru. Dostupné z WWW: <<http://matlab.feld.cvut.cz/view.php?navezclanku=schematicke-znazorneni-parametru-pro-vypocet-r-faktoru&cislocclanku=2004101505>>.
- [17] MEGGELEN, J.V, SMITH, J.Madsen, L. Asterisk™. The Future of Telephony. O'Reilly Media, Inc., Sebastopol 2005. ISBN 0-596-00962-3
- [18] MĚŘÍNSKÝ, Miroslav. Měření MOS-LQO v reálném čase. Praha, 2008. 79 s. Diplomová práce. České vysoké učení technické, Fakulta elektrotechniky.

- [19] Netrino [online]. 2009 [cit. 2010-12-14]. Embedded Systems Glossary. Dostupné z WWW: <<http://www.netrino.com/Embedded-Systems/Glossary>>.
- [20] OmniPeek Network Analyzer [online]. 2011 [cit. 2011-05-09]. WildPackets. Dostupné z WWW: <http://www.wildpackets.com/products/network_analysis_and_monitoring/omnipeek_network_analyzer>.
- [21] Opie handhelds [online]. 2010 [cit. 2010-12-14]. About Opie. Dostupné z WWW: <<http://opie.handhelds.org/>>.
- [22] OPTICOM - The perceptual quality experts [online]. 2010 [cit. 2010-12-14]. PSQM P.861 P861 Perceptual Speech Quality Measure. Dostupné z WWW: <<http://www.psqm.com/>>.
- [23] PECINOVSKÝ, Rudolf. Myslíme objektivě v jazyku Java: Kompletní učebnice pro začátečníky. 2. aktualizované a rozšířené vydání. Praha : Grada Publishing a. s., 2009. 576 s. ISBN 978-80-247-2653-3.
- [24] PISKAČ, Richard. Měření kvality hovoru v IP telefonii. Praha, 2009. 96 s. Diplomová práce. České vysoké učení technické, Fakulta elektrotechniky.
- [25] POSUZOVÁNÍ KVALITY HLASU : POSUZOVÁNÍ KVALITY HLASU. In BRADA, Miroslav; ZELENKA, Jan. Teorie a praxe IP telefonie - 3. dvoudenní odborný seminář Kongresové centrum Hotelu Olšanka, 5. a 6. listopadu 2008. Praha : [s.n.], 2008. s. 10.
- [26] RIŠICA, Matúš. Metodika merania QOS v IP sieťach. Žilina, 2005. 35 s. Bakalářská práce. Žilinská univerzita v Žiline fakulta riadenia a informatiky.
- [27] SLAVATA, Oldřich. Neintruzivní měření kvality přenosu hlasu v telekomunikačních sítích. Praha, 2010. 48 s. Diplomová práce. České vysoké učení technické, Fakulta elektrotechniky.
- [28] SourceForge.net - Find, create, and publish Open Source software for free [online]. 2007, 2009 [cit. 2011-05-09]. WANem - The Wide Area Network emulator. Dostupné z WWW: <<http://wanem.sourceforge.net/>>.
- [29] TCPDUMP/LIBPCAP public repository [online]. 2010 [cit. 2011-05-09]. Programming with pcap. Dostupné z WWW: <<http://www.tcpdump.org/pcap.html>>.
- [30] Telecommunications industry association. TSB-116-A : Telecommunications - IP Telephony Equipment – Voice Quality Recommendations for IP Telephony. In TIA Telecommunications System Bulletin. [s.l.] : [s.n.], 2006. s. 64.
- [31] The GNU Operating System [online]. 1998, 2008 [cit. 2010-11-09]. Announcing ncurses 5.9. Dostupné z WWW: <<http://www.gnu.org/software/ncurses/ncurses.html>>.
- [32] TOULA, Michal. Subjektivní a objektivní měření srozumitelnosti řeči přenesené telekomunikačním kanálem. Praha, 2009. 34 s. Bakalářská práce. ČVUT FEL.
- [33] VEJMELKA, Martin. Moon FEL [online]. 2009 [cit. 2010-12-14]. Programovací jazyk PHP. Dostupné z WWW: <<http://moon.felk.cvut.cz/~vejmem3/oldweb/it/php/php.pdf>>.

- [34] VODRÁŽKA, Jiří Teorie a praxe IP telefonie : Kvalita hovorového signálu v kombinovaných telefonních sítích. In Kvalita hovorového signálu v kombinovaných telefonních sítích. Praha : [s.n.], 2006. s. 14.
- [35] Voip-info.org [online]. 2010 [cit. 2010-12-14]. Asterisk. Dostupné z WWW: <<http://www.voip-info.org/wiki/view/Asterisk>>.
- [36] Voip-info.org [online]. 2010 [cit. 2010-12-14]. CDR. Dostupné z WWW: <<http://www.voip-info.org/wiki/view/Asterisk+config+cdr.conf>>.
- [37] Voip-info.org [online]. 2010 [cit. 2010-12-14]. CDR_CSV. Dostupné z WWW: <<http://www.voip-info.org/wiki/view/Asterisk+cdr+csv>>.
- [38] Voip-info.org [online]. 2010 [cit. 2010-12-14]. CLI. Dostupné z WWW: <<http://www.voip-info.org/wiki/view/Asterisk+CLI>>.
- [39] Voip-info.org [online]. 2010 [cit. 2010-12-14]. Extensions.conf. Dostupné z WWW: <<http://www.voip-info.org/tiki-index.php?page=Asterisk%20config%20extensions.conf>>.
- [40] Voip-info.org [online]. 2010 [cit. 2010-12-14]. Sip.conf. Dostupné z WWW: <<http://www.voip-info.org/wiki/view/Asterisk+config+sip.conf>>.
- [41] VOIPAC Technologies [online]. 2010 [cit. 2010-12-14]. PXA270M. Dostupné z WWW: <<http://www.voipac.com/#27M-DMM-356>>.

SEZNAM POUŽITÝCH ZKRATEK

3SQM – Single Sided Speech Quality Measure
ACR – Absolute Category Rating
CCI – Call Clarity Index
CCR – Comparison Category Rating
CDR – Call Detail Records
CLI – Command Interface Line
DCR – Degradation Category Rating
GoB – Good or Better
INMD – In-Service Non-intrusive Measurement Device
ITU-T – International Telecommunication Union – Telecommunication Standardization Sector
IP – Internet Protocol
LPC – Linear Predictive Coding
MOS – Mean Opinion Score
PAMS – Perceptual Analysis Measurement System
PBX – Private Branch Exchange – pobočková ústředna
PESQ – Perceptual Evaluation of Speech Quality
PoW – Poor or Worse
PSQM – Perceptual Speech Quality Measure
SIP – Session Initiation Protocol
SNR – Signal Noise Ratio – odstup signálu od šumu
TCP – Transmission Control Protocol
UDP – User Datagram Protocol
VOIP – Voice over IP
VM – Virtual Machine

SEZNAM PŘÍLOH

Příloha 1 – Převodní charakteristika mezi MOS a R-faktorem

Příloha 2 – Zjednodušený vývojový diagram

Obsah přiloženého CD

Elektronická verze práce v souboru *xbilek15_dp.pdf*.

Úplný vývojový diagram v souboru *vyvojovy_diagram.pdf*.

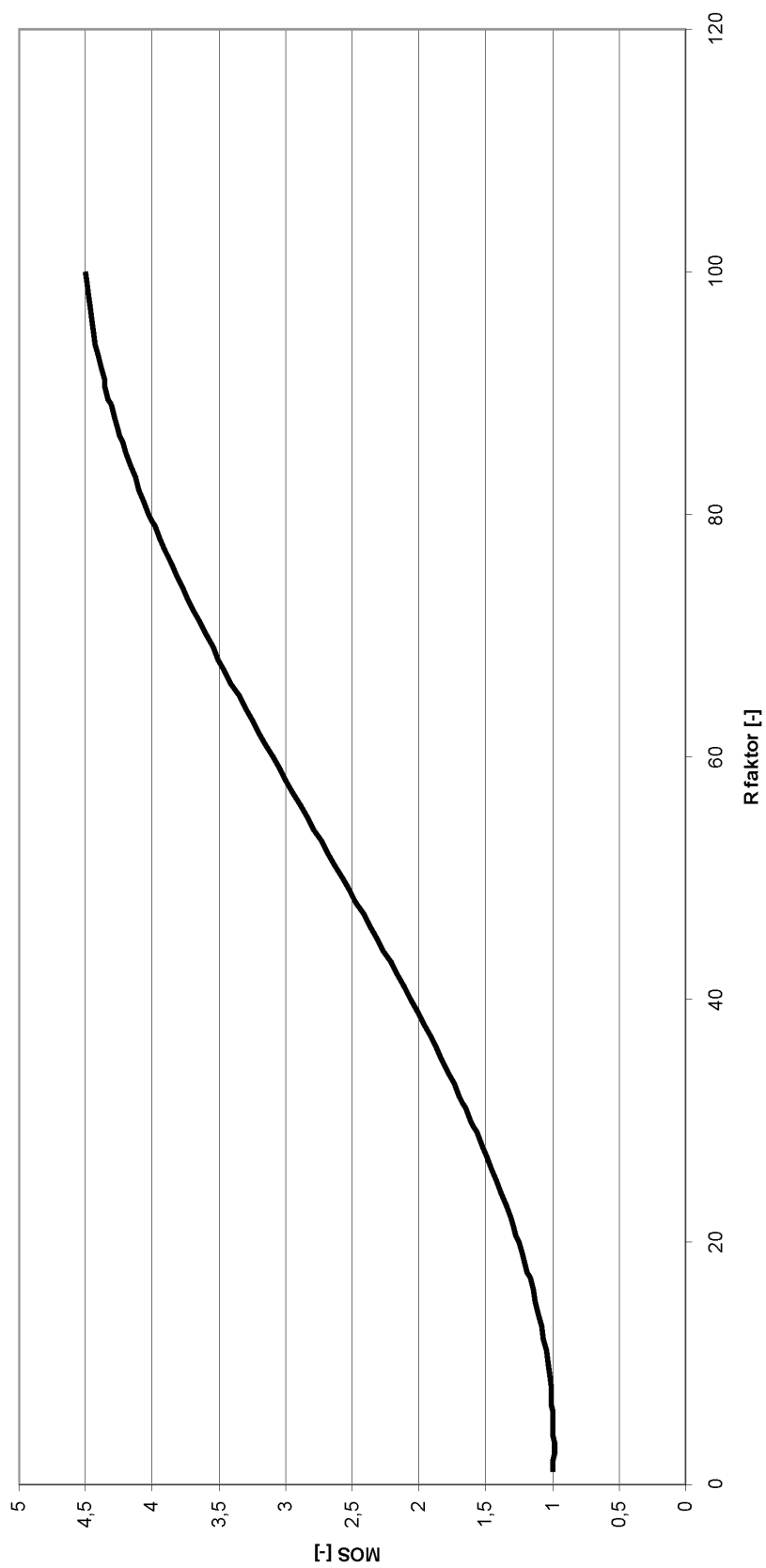
Aplikace měření kvality ve verzi pro PC v adresáři *Aplikace_mereni_kvality_PC*.

Aplikace měření kvality ve verzi pro ARM v adresáři *Aplikace_mereni_kvality_ARM*.

Návod k překladu PBX Asterisk pro vývojový kit v souboru *Asterisk_install_ARM.txt*

PŘÍLOHA 1 – PŘEVODNÍ CHARAKTERISTIKA MEZI MOS A R-FAKTOREM

Převodní vztah mezi R faktorem a MOS



PŘÍLOHA 2 – ZJEDNODUŠENÝ VÝVOJOVÝ DIAGRAM

