

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

Diplomová práce

Embedded Asterisk PBX

2012

Josef Stoklasa

Zadání diplomové práce

Student: **Bc. Josef Stoklasa**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2601T013 Telekomunikační technika

Téma: **Embedded Asterisk PBX**
Embedded Asterisk PBX

Zásady pro vypracování:

Softwarová PBX Asterisk je velmi rošířená aplikace, která ovšem nese sebou i nezanedbatelné náklady týkající se vlastního provozu. Cílem diplomové práce je navrhnout a prakticky realizovat SOHO (Small Office/Home Office) řešení, které by splňovalo nároky domácích uživatelů a malých firem, a zároveň by nebylo energeticky náročné. Diplomant bude pracovat na zařízení PC Engine Alix a bude používat linuxovou distribuci Voyage anebo openWRT. Výstupem diplomové práce bude Embedded Asterisk PBX s nízkou energetickou náročností, zařízení bude vhodné pro SOHO segmentu trhu.

1. Projekt softvérového komunikačního systému Asterisk.
2. HW platforma ALIX a green computing.
3. Implementace Asterisku na HW ALIX.
4. Posouzení energetické náročnosti a porovnání se současnými systémy.
5. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

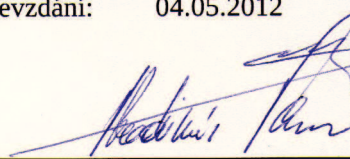
1. Meggelen J., Smith J., Madsen L. *Asterisk: The Future of Telephony*. O'Reilly Media; 2nd edition, 2007, ISBN 978-0596510480
2. Vozňák, M. *Voice over IP*. Vysokoškolská skripta, 176 stran. Vydavatel: VŠB-TU Ostrava, 1. vydání, v Ostravě, 2008, ISBN 978-8024818283

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

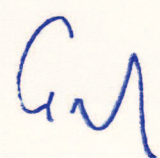
Vedoucí diplomové práce: **doc. Ing. Miroslav Vozňák, Ph.D.**

Datum zadání: 19.11.2010

Datum odevzdání: 04.05.2012


prof. RNDr. Vladimír Vašínek, CSc.
vedoucí katedry




prof. RNDr. Václav Snášel, CSc.
děkan fakulty

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

V Ostravě 2. května 2012



.....

Rád bych na tomto místě poděkoval panu Doc.Ing. Miroslavu Vozňákovi Ph.D. za odborné konzultace, bez nichž by tato práce nikdy nevznikla a také Ing. Janu Rozhonovi za asistenci při zátěžových testech Asterisk PBX.

Abstrakt

VoIP telefonie je nyní v dobách expanze a je stále častěji používanou technologií pro realizaci firemní, ale i soukromé telefonie. Softwarová pobočková ústředna (dále jen PBX) na bázi Asterisk je dnes velmi oblíbenou variantou pro řešení firemní telefonie a to především díky velmi nízkým nákladům na zřízení. Nese však s sebou nemalé náklady na provoz a to hlavně co do spotřeby elektrické energie. Náklady jsou tedy přímo úměrné elektrickému příkonu daného hardware na němž Asterisk PBX provozujeme. Tato Diplomová práce pojednává o minimalizaci nákladů na provoz PBX s ohledem na co možná nejmenší ztrátu výpočetního výkonu a zároveň možnosti rozšiřitelnosti takového ústředny jako celku v komplexní síťový miniserver, jenž je dále nazýván jako HomeRouter. Cílem této práce je navrhnout a zrealizovat ucelený minimalistický systém zvládající všechny běžné funkce softwarové PBX, přičemž je kladen důraz právě na minimální energetickou náročnost. Takováto zařízení se dnes běžně nazývají termínem SOHO (Small Office/Home Office). V poslední části této práce budou také zmíněny možné rozšíření uceleného minimalistického systému o další serverové aplikace užívané miniserverem HomeRouter jako je HTTP server, FTP server, SMB server, VPN koncentrátor a další.

Klíčová slova: SOHO, Asterisk PBX, VoIP telefonie, Alix.2D13, Voyage Linux, protokol SIP, x86 architektura, Ethernet, Wi-fi, QoS,

Abstract

VoIP is currently at the time of expansion and is still more frequently used technology for real-time communication. Software branch exchange (PBX) based on Asterisk is very popular solution for corporate telephony at this time. Because Asterisk is open-source software, in that case gratis, therefor is his running related only with energy cost. Energy cost in this case is direct proportional for power consumption of hardware where Asterisk PBX is running. Therefor this theses is focused on minimalisation of energy cost needed for Asterisk PBX running with reference to minimal lost of computational power. Also is this project focused on chances of additional network services running also on this PBX, and thus create complex mini-server which is further named as HomeRouter. Specific of this project is emphasisation for minimal energy cost related with network services running on specified hardware. This devices are usually named as SOHO (Small Office/Home Office). In last section of this theses is also described howto implement

additional network services running on HomeRouter as HTTP, FTP, SMB server, VPN concentrator and so on.

Keywords: SOHO, Asterisk PBX, VoIP telephony, Alix.2D13, Voyage Linux, protocol SIP, x86 architecture, Ethernet, Wi-fi, QoS,

Seznam použitých zkratk a symbolů

ACD	– Automatic Call Distribution
ATM	– Asynchronous Transfer Mode
DiffServ	– Differentiated Services
DTMF	– Dual-Tone Multi-Frequency
FTP	– File Transfer Protocol
GNU	– GNU's Not Unix
GPL	– General Public Licence
HTML	– Hyper Text Markup Language
HTTP	– Hyper Text Transfer Protocol
IntServ	– Integrated Services
ISP	– Internet Service Provider
IVR	– Interactive Voice Response
NAT	– Network Address Translate
PBX	– Private Branch eXchange
QoS	– Quality of Service
RTD	– Round-Trip Delay
SIP	– Session Initiation Protocol
SMB	– Server Message Block
SOHO	– Small Office/Home Office
SSH	– Secure Shell
SSL	– Secure Socket Layer
VoIP	– Voice over Internet Protocol
VPN	– Virtual Private Network
Wi-fi	– Wireless Fidelity
WINS	– Windows Internet Naming Service

Obsah

1	Úvod	6
1.1	Motivace	6
1.2	Náplň práce	6
2	Projekt Asterisk	7
2.1	Doplňové služby Asterisku	8
2.2	Popis podporovaných rozhraní, kanálů a kodeků v Asterisk	11
2.3	Architektura Asterisku	16
2.4	Priorizace toku dat, metody obsluhy paketových front a podpora značkování provozu v Asterisku	17
3	HW platforma ALIX a green computing	26
3.1	Platforma ALIX	26
3.2	Green Computing	33
4	Implementace Asterisku na HW ALIX	37
4.1	Voyage Linux	37
4.2	Instalace Voyage Linux na ALIX.2D13	38
4.3	Instalace a nastavení Asterisku	41
5	Posouzení energetické náročnosti a porovnání se současnými systémy	47
5.1	Porovnání spotřeby vzorové PC sestavy a desky ALIX.2D13	47
5.2	Kalkulace návratnosti investice do Embedded Asterisk PBX	48
6	Zhodnocení dosažených výsledků	50
6.1	Měření datové propustnosti Embedded Asterisk PBX	50
6.2	Měření počtu současně obslužených SIP hovorů	51
6.3	Možnosti rozšíření Embedded Asterisk PBX na miniserver HomeRouter	55
7	Závěr	60
8	Reference	61
	Přílohy	64
A	Konfigurace a návody	65
A.1	Postup pro instalaci voayge linux z LiveCD	65
A.2	Nastavení souboru <i>sip.conf</i> pro testování pomocí SIPP	66
A.3	Soubor <i>extensions.conf</i> pro testování pomocí SIPP bez současného nahrávání hovorů	67
A.4	Soubor <i>extensions.conf</i> pro testování pomocí SIPP se současným nahráváním hovorů	67
A.5	Konfigurace <i>extensions.conf</i> pro funkci podmíněného nahrávání	68

A.6	Nastavení souborů <i>interfaces</i> a <i>hostapd.wlan0.conf</i> pro vytvoření AP	68
A.7	Nastavení souboru <i>dnsmasq.conf</i> pro funkci DHCP serveru	69
A.8	Nastavení souboru <i>vsftpd.conf</i> pro funkci FTP serveru	69
A.9	Nastavení souboru <i>lighttpd.conf</i> pro funkci HTTP serveru	70
A.10	Nastavení souboru <i>smb.conf</i> pro sdílení v sítích Windows	70
A.11	Nastavení souboru <i>config.xml</i> pro sdílení multimédií pomocí UPnP/DLNA	71
A.12	Nastavení souboru <i>mpd.conf</i> pro funkci síťového přehrávače hudby	74
A.13	Nastavení konfiguračních souborů pro funkci VPN koncentrátoru	74

Seznam tabulek

1	Srovnání kodeků podle MOS a MIPS	15
2	Srovnání parametrů nejpoužívanějších kodeků	15
3	Doporučené značkování provozu dle IEEE 802.1Q-2005.	19
4	Doporučené hodnoty DSCP pro Assured Forwarding	19
5	Kanálové ovladače podporující QoS značkování	24
6	Soubory podporující nastavení pole ToS pro signalizaci	24
7	Doporučené hodnoty ToS a CoS v Asterisku	24
8	Modely desek ALIX	27
9	Čipsety podporované pro použití s tinyBIOS	29
10	Spotřeba průměrné PC sestavy při plném zatížení	47
11	Spotřeba měřené PC sestavy při 20% zatížení CPU	47
12	Spotřeba Embedded Asterisk PBX při 20% zatížení CPU	48
13	Ceny komponent potřebných k Embedded Asterisk PBX	48
14	Porovnání finanční náročnosti na spotřebovanou el. energii . .	49
15	Připojené stanice pro propustnosti Embedded Asterisk PBX . . .	50

Seznam obrázků

1	Blokové schéma architektury Asterisku (použito se souhlasem autora [3])	16
2	Vliv variability zpoždění - jitter	17
3	Srovnání rámců ISL a 802.1Q (použito se souhlasem autora [10])	18
4	IP záhlaví s použitím a bez použití DiffServ (použito se souhlasem autora [10])	20
5	Princip obsluhy paketových front metodou MDRR (použito se souhlasem autora [11])	21
6	Princip obsluhy paketových front metodou WFQ (použito se souhlasem autora [11])	23
7	AMD Geode LX 800 (500MHz) CPU	30
8	Blokové schéma architektury AMD Geode LX [21]	32
9	Test výkonosti AMD Geode LX800 uváděn výrobcem [21]	34
10	ALIX.2D13	35
11	Logo Voyage Linux	38
12	Úvodní obrazovka LiveCD Voyage Linux v emulátoru Qemu	40
13	Grafické rozhraní Asterisku	42
14	Výsledek práce - Embedded Asterisk PBX	49
15	Testování Asterisk PBX nástrojem SIPP	52
16	Graf vytížení CPU v závislosti na počtu současných hovorů	53
17	Vytížení CPU procesem <i>asterisk</i> při 40-ti současných hovorech bez a s nahráváním hovorů	55
18	GMPC - Klient pro ovládání MPD serveru	58

Seznam výpisů zdrojového kódu

1	Příkaz pro mapování Linux socket priority na 802.1p CoS	24
2	Parametry v konfiguračním souboru <i>iax.conf</i> pro nastavení ToS a CoS . . .	25
3	Parametry v konfiguračním souboru <i>sip.conf</i> pro nastavení ToS a CoS . . .	25
4	Instalace virtuálního stroje Qemu	39
5	Stažení poslední verze Voyage MPD přes příkazovou řádku	39
6	Kontrola identifikátoru disku	39
7	Spuštění Qemu pro zavedení Voyage Linux ze staženého LiveCD	39
8	Instalace Voyage Linuxu na CF kartu - krok 1	40
9	Instalace Voyage Linuxu na CF kartu - krok 2	41
10	Instalace Voyage Linuxu na CF kartu - krok 3	41
11	Ověření korektního zavádění systému	41
12	Instalace Asterisku ve Voyage Linux	41
13	Instalace grafického rozhraní pro Asterisk	42
14	Nastavení souboru <i>http.conf</i> potřebná pro správnou funkci Asterisk GUI .	43
15	Nastavení v souboru <i>manager.conf</i> potřebná pro správnou funkci Asterisk GUI	43
16	Nastavení v souboru <i>sip.conf</i> potřebná pro základní zprovoznění SIP telefonie	44
17	Nastavení v souboru <i>extensions.conf</i> potřebná pro základní zprovoznění SIP telefonie	44
18	Nastavení monitorování hovoru v souboru <i>extensions.conf</i>	45
19	Měření TCP propustnosti v rámci jedné podsítě	51
20	Měření TCP propustnosti mezi dvěma podsítěmi	51
21	Měření CPU nástrojem <i>top</i>	53
22	Zvýšení priority procesu <i>asterisk</i>	54
23	Bezdrátové rozhraní <i>wlan0</i> po zapojení karty CM9	56

1 Úvod

1.1 Motivace

Žijeme v době kdy internetová komunikace vládne světu. Na dnešním telekomunikačním trhu se pohybuje nemálo aplikací umožňující jakoukoliv komunikaci v reálném čase, ať už komunikaci textovou, hlasovou či komunikaci s možností videopřenosu. Nejrychlejším a zpravidla nejvíce používaným způsobem komunikace ve firemní sféře však stále zůstává komunikace hlasová a to především díky mobilním telefonům. Denně zaměstnanci firem protелефonují stovky minut, za něž firma platí operátorům nemalé peníze. Zaměříme-li se na odvětví pevné telefonní sítě, můžeme s využitím VoIP telefonie ušetřit i více než dvě třetiny nákladů za telefonní účty. Proto je také potřeba zohlednit náklady na provoz takto řešené telefonie a hledat nejschůdnější řešení. Nemusíme však zůstat jen u odvětví pevné telefonní sítě, jelikož naprostá většina dnes dostupných tzv. chytrých telefonů (SmartPhone) má obvykle VoIP klienta instalovaného buďto nativně, či případně existují dodatečné aplikace umožňující tento způsob telefonování v mobilním telefonu. Z výše uvedeného tedy vyplývá, že hlavní myšlenkou této práce je navrhnout a zrealizovat komplexní minimalistický systém nahrazující dnes běžně používané systémy, avšak s řádově několikanásobně menší spotřebou elektrické energie.

1.2 Náplň práce

Jak již bylo předesláno, tato práce pojednává o vytvoření minimalistického systému provozujícího Embedded Asterisk PBX. Tento Embedded Asterisk PBX realizuje softwarovou PBX, přičemž jej lze následně rozšířit také o další síťové služby jako VPN koncentrátor, SMB server, FTP server atd. Tento minimalizovaný server provozující několik síťových služeb současně je potom ve zbytku práce, resp. v kapitole 6.3 nazýván HomeRouter. Důraz je zde kladen na minimalizaci provozních nákladů, přičemž je optimalizován tak aby byl jeho výpočetní výkon pro požadované služby dostatečný a mohl tedy v SOHO segmentu trhu nahradit běžně používané PC systémy. Výpočetní výkon těchto běžně užívaných PC systémů je mnohdy využit jen z nepatrné části, avšak jejich spotřeba elektrické energie se může pohybovat v řádech pěti až deseti násobku spotřeby elektrické energie řešení HomeRouter, jak je popsáno v kapitole 5.

2 Projekt Asterisk

Projekt Asterisk byl napsán roku 1999 Markem Spencerem, který je mimo jiné také autorem projektů Pidgin messenger, či L2TP daemon. Původně vznikl jako nadšenecký projekt, jehož záměrem zpočátku nebylo vytvořit robustní a komplexní open-source PBX úspěšně konkurující komerčním řešením. Jak uvádí některé prameny [13] při vzniku Asterisku byl Mark Spencer čerstvým absolventem Auburn University v Alabamě, a tehdy se rozhodl napsat vlastní PBX namísto zakoupení komerčního řešení na které údajně nebylo dost prostředků. Marku Spencerovi se později podařilo najít kolegu jménem Jim Dixon jenž naopak vyvíjel open-source hardware [4]. Jejich prvním společným krokem bylo vytvoření open-source T1 karty. Podařilo se jim tedy společně položit základní stavební kámen pro budoucí open-source PBX nesoucí jméno Asterisk. Mark Spencer je později zaožil firmu Digium, která dodává hardware určen přímo pro Asterisk (např. T1/E1 /FXO/FXS) a také financuje jeho vývoj. Asterisk si postupem času získal takovou oblibu, že je v dnešní době bezesporu nejoblíbenějším řešením VoIP telefonie v kategorii open-source. Co do podpory různorodosti komunikačních hlasových protokolů nabízí velice širokou škálu. Podporuje protokoly SIP, IAX (vytvořen speciálně pro Asterisk), MGCP, H.323 i Cisco proprietární protokol Skinny (SCCP). Mezi těmito protokoly je dnes nejhojněji používaným protokolem protokol SIP, který je výrobcí hardwaru pro VoIP oblíben především pro možnost jednoduché implementace. V těsném závěsu pak, především v korporátních prostředích, nalezneme Cisco proprietární hlasový protokol Skinny. Projekt Asterisk ať už jako celek, nebo jeho části, lze provozovat pod licencí GNU General Public License (GPL) a to na platformách operačních systémů Linux a Unix. Může plnit různé funkce v dané VoIP infrastruktuře, přičemž je používán především k následujícím účelům:

- VoIP gateway pro protokoly MGCP, SIP, IAX, H.323, nebo gateway do PSTN
- Pobočková ústředna (PBX)
- Voicemail služby s adresářem
- IVR server
- Softwarová ústředna (Softswitch)
- Konferenční server
- Packet voice server
- Šifrování telefonních nebo faxových volání
- Překlad čísel
- Aplikace Calling card
- Prediktivní volič (Predictive dialer)

- Řazení volání do front se vzdáleným zprostředkovatelem
- Vzdálené „kanceláře“ pro existující PBX

Projekt Asterisk pak tedy může snadno najít uplatnění v Call centrech, veřejné síti Internet u jednotlivých ISP poskytujících VoIP služby, ve firmách všech velikostí jako hlavní PBX, či gateway do PSTN, a v neposlední řadě také jako PBX pro odlehlá firemní pracoviště, či kanceláře. Právě na poslední ze zmíněných variant je zaměřena tato diplomová práce, přičemž klade důraz na minimalizaci s sebou nesoucích nákladů.

2.1 Doplněné služby Asterisku

Asterisk nabízí nesčetné množství doplňkových služeb klasických i pokročilých, které jsou obvykle poskytovány pouze nákladnými komerčními PBX. Jako doplňkové služby a funkce nabízí Asterisk například následující [13]:

- ACCOUNT CODE, používá se pro účely tarifování. Volající před volbou čísla vloží svůj kód. Do CDR (Call Detail Recording) se zaznamenávají údaje o délce hovoru, volaném čísle, ceně, atd...
- AUTOMATED ATTENDANT, volající je přepojen na požadovanou pobočku bez účasti spojovatelky.
- AUTOMATIC HOLD, pokud chceme provést druhý hovor, tak můžeme inicializovat bez zavěšení předchozího a Asterisk první hovor automaticky podrží, přičemž se k němu můžeme zpětně vrátit.
- BLACKLIST, seznam nežádoucích čísel, které jsou jako příchozí hovory odmítnuty.
- CALL TRANSFER, jedná se o předání hovoru.
- BLIND TRANSFER, je předání hovoru na jinou pobočku bez sledování toho, zda hovor někdo přijme, hovor je předán i s vyzváněním.
- CALL DETAIL RECORD, záznam hovorů uskutečněných v PBX, který obsahuje volající číslo, volané číslo, datum, délku hovoru a případně další informace.
- CALL FORWARDING ON BUSY, příchozí hovor je automaticky přeměřován za podmínky, že je volané číslo obsazeno.
- CALL FORWARDING ON NO ANSWER, příchozí hovor je automaticky přeměřován jen tehdy, pokud volané číslo neodpovídá, tzn. po určitém čase.
- CALL FORWARDING UNCONDITIONALLY, jedná se o okamžité přeměřování bez podmínek.
- CALL MONITORING, evidence příchozích, odchozích a zmeškaných volání, přístup přes web, uživatel po autentizaci vidí seznam volání, které se týkají jeho účtu.

-
- CALL PARKING, odložení hovoru do virtuálního úložiště s možností jeho vyzvednutí stejnou nebo jinou pobočkou.
 - CALL QUEUING, umožňuje řadit příchozí hovory do fronty a vyzvedávat je dalším volným účastníkem konkrétní skupiny.
 - CALL RECORDING, umožňuje zaznamenávat hovory, zaznamenané hovory jsou uloženy v požadovaném formátu (např. PCM či GSM) a nabídnuty k přehrání oprávněnému uživateli, přihlášení probíhá přes https a po zadání hesla jsou nabídnuty pouze hovory týkající se konkrétní pobočky autentizovaného uživatele.
 - CALL RETRIEVAL, funkce vyvolá osobu, která může převzít volání.
 - CALL ROUTING, je provolení na pobočku (DDI – Direct Dialing In, provolba).
 - CALL SNOOPING, umožňuje odposlouchávat určenou skupinu telefonů.
 - CALLER ID, je funkce zobrazení čísla volajícího a jména volajícího.
 - CALLER ID BLOCKING, hovor je odmítnut na základě identifikace volajícího.
 - CALL WAITING, je upozornění na čekající volání během sestaveného spojení, po jeho přijetí je možné střídání mezi oběma hovory.
 - CALL ID ON CALL WAITING, je identifikace dalšího volajícího při probíhajícím hovoru.
 - CONFERENCE BRIDGING. vytvoří konferenci mezi terminály různých typů jako lokální pobočkou, vzdálenou linkou, mobilním účastníkem, VoIP spojením, apod...
 - DATABASE STORE / RETRIEVAL, ukládá informace o hovorech do DB pro pozdější využití.
 - DATABASE INTEGRATION, Asterisk umožňuje poskytování informací o volajícím účastníkovi volanému před přijmutím volání nebo během hovoru.
 - DIAL BY NAME, namísto čísla je možné volit i jméno (jako alias).
 - DISTINCTIVE RING, jedná se o rozdílný typ vyzvánění založený na identifikaci volajícího.
 - DUNDI (Distributed Universal Number Discovery), je distribuovaný systém směrování, který v síti Asterisků umožní jednak rozložení zátěže mezi různé servery a jednak zvýšení odolnosti při výpadku některého z Asterisk serverů (cluster – několik Asterisků, které se navenek tváří jako jeden velký softswitch).
 - DO NOT DISTURB, aktivací funkce nerušit je volání přesměrováno na ohlášení, spojovatelku nebo jinou pobočku, apod...

-
- ENUM, Asterisk podporuje vyhledávání telefonních čísel přes DNS, kde je realizováno mapování telefonních čísel na jmenné identifikátory (URI), pokud je spojení na vyhledanou URI adresu nedostupné, tak se použije další pravidlo (např. směrování přes PSTN).
 - INTERACTIVE DIRECTORY LISTING, umožňuje volajícímu účastníkovi interaktivní vyhledání volaného podle jeho jména v korporátním adresáři.
 - INTERACTIVE VOICE RESPONSE, IVR je pokročilý systém pro obsluhu příchozích volání, volající prochází hlasovým menu a pomocí volby čísel volí možnosti spojení.
 - LOCAL AND REMOTE CALL AGENTS, účastník se může pomocí své identifikace přihlásit na kterýkoliv telefonu a používat ho jako svůj vlastní (tzn. s vlastním číslem, nastavením služeb, atd.)
 - MUSIC ON HOLD, hudba na přidržené lince, přičemž audio soubory lze vytvářet jednoduchým způsobem.
 - PREDICTIVE DIALER, funkce je používána odchozími centry volání, spojení se sestavuje na základě statistického modelu, který určuje, kdy bude volaná strana dostupná.
 - PRIVACY MANAGER, jestliže je kód pro vzdálený přístup zablokován (viz. LOCAL AND REMOTE CALL AGENTS), potom ručním zadáním čísla Privacy Manager zkontroluje, zda je číslo na Black listu nebo White listu a podle toho volbu povolí nebo zamítne.
 - PROTOCOL CONVERSION, umožňuje spojení mezi sítěmi používajícími rozdílné protokoly.
 - REMOTE CALL PICKUP, umožňuje vyzvednout hovor, který vyzvání na jiné pobočce.
 - REMOTE OFFICE SUPPORT, umožňuje přihlásit telefon z jiné PBX tak, že má vlastnosti lokální pobočky.
 - ROAMING EXTENSIONS, jednotlivé osoby jsou vybaveny číslem pobočky a kódem, pomocí kterého se mohou přihlásit na kterémkoliv pobočkovém telefonu, tato služba je odlišná od LOCAL AND REMOTE CALL AGENTS tím, že číslo pobočky, pokud se zrovna nepoužívá, neexistuje v Dialplan.
 - VOICEMAIL, umožňuje nahrát vzkaz pro volaného, zpřístupnit nahrané vzkazy z telefonu, přes web anebo odeslat vzkaz do poštovní schránky uživatele jako email.
 - ROUTE BY CALLER ID, hovor je směrován na základě čísla volajícího na pobočku, do fronty nebo do skupiny účastníků (Ring Group).

- SMS MESSAGING, Asterisk umožňuje pomocí SMS upozorňovat např. zmeškaná volání a zanechané vzkazy, SMS se posílají přes SMS bránu (může to být GSM modem lokálně připojený k Asterisku).
- SPELL/SAY, funkce umožňuje přečíst text, např. email.
- SUPERVISED TRANSFER, je předání volání řízené automatickým zařízením (např. Voice Response Unit), které vyhodnotí výsledek předání – přijato, obsazeno, nevyzvednuto.
- TALK DETECTION, funkce umí detekovat hovor (rozezná záznamník od osoby).
- THREE-WAY CALLING, je konference tří účastníků, je možné ovšem dělat i konference o několika desítkách účastníků (na Asterisku otestováno do 30-ti) s využitím konferenční místnosti (Meet-Me).
- TIME AND DATE, funkce čte čas a datum volajícímu.
- TRANSCODING, Asterisk umožňuje konverzi mezi různými kodeky.
- TRUNKING, je funkce připojení do klasické telefonní sítě pomocí interní karty v Asterisku.

Nejnovější stabilní verze 1.8 je doplněna například o funkce jako Podpora spolupráce s kalendáři, Integrace Jabber/XMPP funkcí, Call Event Logging (CEL), Call Completion Supplementary Services (CCSS), Advice of Charge (AOC-S, AOC-D, AOC-E), zlepšená podpora TDM a T.38 faxování atd. [14]

2.2 Popis podporovaných rozhraní, kanálů a kodeků v Asterisk

Asterisk je od základu navržen jako modulární systém který nabízí možnost snadné rozšiřitelnosti. Umožňuje implementovat nové technologie, přičemž je jeho cílem podporovat nejen všechny aktuálně dostupné, ale i budoucí technologie používané v prostředí telefonie. Použitelná rozhraní jsou rozdělena do tří základních skupin[1]:

- Zaptel hardware
- non-Zaptel hardware
- packet voice

2.2.1 Zaptel hardware

Zaptel hardware vznikl jako open-source alternativa k tehdy dostupnému hardware TDM, přičemž jeho tvorba nebyla vůbec jednoduchým úkolem. Běžně dostupný TDM hardware (např. Dialogic, později majetkem Intelu) byl totiž patentován a navíc byl příliš

drahý. Pro dosažení vymezeného cíle bylo přistoupeno ke zcela inovativní metodě. Namísto zpracovávání TDM hardwarově, byl přidán hostitelský procesor a Asterisk pracoval výhradně s tímto procesorem. Jak se postupem času stávaly CPU stále rychlejšími a rychlejšími, začalo být rozumnější pro toto TDM zpracování ponechat software využívat hlavní CPU počítače. Po přidání TDM podpory do Asterisku začala s revolučními změnami v oblasti telefonie firma *Zapata Telephony* a to výrobou pseudo TDM rozhraní, které nazvala Zaptel. Inspirací pro Zapata Telephony se stal mexický revolucionář generál Emiliano Zapata, jelikož firma Zapata se stejně jako on rozhodla provést revoluční změny. Pseudo-TDM architektura poskytuje téměř stejnou kvalitu jako klasické TDM a real-time schopnosti jakou má hardware TDM. Podstatným rozdílem je však podstatně nižší cena a vyšší flexibilita. Zaptel rozhraní dodává např. firma Digium (hradí vývoj Asterisku) a to pro různé varianty síťových rozhraní včetně PSTN, POTS, T1, E1, PRI, PRA, EM a mnoho dalších.[1]

2.2.2 Non-Zaptel hardware

Následující rozhraní poskytnou možnost propojení s ostatními telefonními službami, nepodporují ale pseudo-TDM komutování [2]:

- ISDN4Linux - základní ISDN rozhraní pro Linux
- OSS/Alsa - rozhraní zvukové karty
- Linux Telephony Interface (LTI)
- Phonejack/Linejack
- Dialogic hardware - standardně není Asteriskem podporován, ale je k dispozici za poplatek a to jako doplněk pro zákazníky s hardwarem Intel/Dialogic

2.2.3 Packet voice

Zatímco rozhraní Zaptel sloužící pro spojení serveru přímo k PSTN již pracovalo spolehlivě, bylo nutno vyřešit komunikaci čistě přes IP, později Frame Relay. Jednalo se o případ, kdy začátek nebo konec hlasové relace vznikl někde jinde, tak jako v případech standardních protokolů pro komunikaci přes paketové sítě IP a Frame Relay. Zde jsou využívána rozhraní, která nepožadují specializovaný hardware ale pracují s hardware navrženým pro výše zmíněné paketové sítě. Jelikož autor Asterisku Mark Spencer neměl v přílišné oblibě standardizovaný protokol H.323, rozhodl se navrhnout a realizovat svůj vlastní protokol. Výsledkem je protokol IAX (Inter Asterisk eXchange) jenž se stará o signalizaci a transport packet voice mezi dvěma připojenými uzly. Ačkoliv název naznačuje přítomnost Asterisku na obou koncích komunikace, IAX může ve skutečnosti spojit každé dva koncové body podporující tento protokol bez nutné přítomnosti Asterisku. Následně byla přidána podpora součinnosti s ostatními VoIP systémy a podpora pro další packet voice protokoly. Jedná se o protokoly SIP, H.323, MGCP (Media Gateway Control Protocol), VoFR (Voice over Frame Relay) a také SCCP. S takovou podporou se Asterisk stává

v současné době ideální PBX na bázi IP pro prostředí s rozmanitými typy používaných telefonů [2].

2.2.4 Zacházení s kanály v Asterisku

Prostřednictvím kanálů vstupují do systému různé formáty komunikace. Kanály jsou logická spojení se signalizačními a přenosovými cestami, které může Asterisk využívat k vytváření a spojování jednotlivých hovorů. Asterisk nerozlišuje typy kanálu FXO (Foreign eXchange Office) a FXS (Foreign eXchange Station), nerozlišuje tedy mezi telefonními linkami a telefony. Každý hovor je umístěn na odlišném kanále. Asterisk s těmito kanály zachází jako s přípojnými body, jejichž vzájemné působení a pravidla se definují v Dial-plánu *extensions.conf*. Je dobré si uvědomit, že i když se kanály liší v rámci použité technologie a typu rozhraní, Asterisk umožňuje zacházet se všemi shodně. V současnosti například existuje několik nezávislých implementací kanálu H.323 do systému Asterisk (*h323*, *oh323*, *ooh323*). Vlastní kanál je použit pochopitelně i pro SIP a IAX.

Konfigurace kanálů pro protokol IAX se provádí modifikací souboru *iax.conf*. Kanál *chan_local* je pseudokanál, jež se používá pro vytvoření smyčky, která volá zpět do Dialplánu v různých kontextech. Užitečné může být například rekurzivní směrování, které je schopno vracet se do Dialplánu po ukončení volání. Kanálový modul pro SIP umožní Asterisku VoIP komunikaci se SIP telefony a ústřednami. Konfigurace SIP kanálů/klientů se provádí modifikací souboru *sip.conf*. ZAP kanálový modul poskytne mezivrstvu (anglicky interface layer) mezi Asteriskem na straně jedné a Zaptel a/nebo ZapHFC ovladači rozhraní na straně druhé. Konfigurace ZAP kanálů provádíme modifikací souboru *zapata.conf*. Dialplan je konfigurován v souboru *extensions.conf*. Jedná se o nejdůležitější konfigurační soubor v celém systému. Pomocí definovaných pravidel řídí způsob ovládání a směrování příchozích a odchozích volání [13].

2.2.5 Podporované kodeky

Asterisk standardně podporuje tyto kodeky:

- ADPCM
- G.711 u-law
- G.711 a-law
- G.723.1
- G.726
- G.729
- GSM
- iLBC (internet Low Bitrate Codec)

- LPC10
- Speex

Z výše uvedených zde popíši nejpoužívanější kodeky používané nejen v Asterisk, ale VoIP telefonii obecně [1]:

- **G.711** je základní PCM kódem, který se využívá také v běžné telefonní síti PSTN a tedy i kvalita hovoru je stejná jako kvalita v sítích PSTN. MOS má hodnotu 4,2, rámeček trvá 0,125 ms;
- **G.723.1** standardně používá buďto kódování MP-MLQ anebo ACELP, přičemž první typ kódování využívá šířku pásma 6,3 kbps a druhý typ 5,3 kbps. Standardní zpoždění u tohoto typu kodeku je 30 ms. MOS je pak 3,9 při použití kódování MP-MLQ a 3,65 při použití ACELP;
- **G.726** kodek používá kódování ADPCM, přičemž potřebná šířka pásma je 16, 24, 32 a 40 kbps. Kodek může zpracovávat bloky různé délky v závislosti na požadovaném zpoždění. Pro 32 kbps se uvádí MOS skóre 3,85;
- **G.728** používá kódování LD-CELP a potřebná šířka pásma kterou tento kodek konzumuje je 16 kbps, přičemž MOS skóre je hodnoceno známkou 3,6;
- **G.729** využívá kódování CS-ACELP (Conjugate Structure Algebraic Code Excited Linear Prediction). Potřebná šířka pásma je 8 kbps, přičemž zpoždění rámečku je 10 ms. MOS je stanoven na 3,92;
- **GSM** konzumuje šířku pásma přibližně 13 kbps a je mnohem rychlejší než metody založené na slovníku CELP. MOS je uváděno 3,5;
- **iLBC** kodek iLBC (internet Low Bitrate Codec) byl vyvinut firmou Global IP Sound pro úzkopásmový hovor. Využívá algoritmu Block Independent Linear Predictive Coding a pracuje s šířkou pásma 13,33 kbps s rámečkem 30 ms nebo 15,2 kbps s rámečkem 20 ms. Pro druhý ze zmíněných případů je uváděno MOS 4,14;

V tabulce 1 můžeme vidět srovnání jednotlivých kodeků řazených dle parametru MOS (Mean Option Score).

Kodek	Typ	Rychlost [kbps]	MOS	MIPS
G.711	PCM	64	4,2	0,1
G.723.1	MP-MLQ	5.33	3,65	16
G.723.1	ACELP	6.4	3,90	16
G.726	ADPCM	32	3,85	12
G.728	LD-CELP	16	3,61	33
G.729	CS-ACELP	8	3,92	20
G.729 x 2 kódování	CS-ACELP	8	3,27	20
G.729 x 3 kódování	CS-ACELP	8	2,68	20

Tabulka 1: Srovnání kodeků podle MOS a MIPS

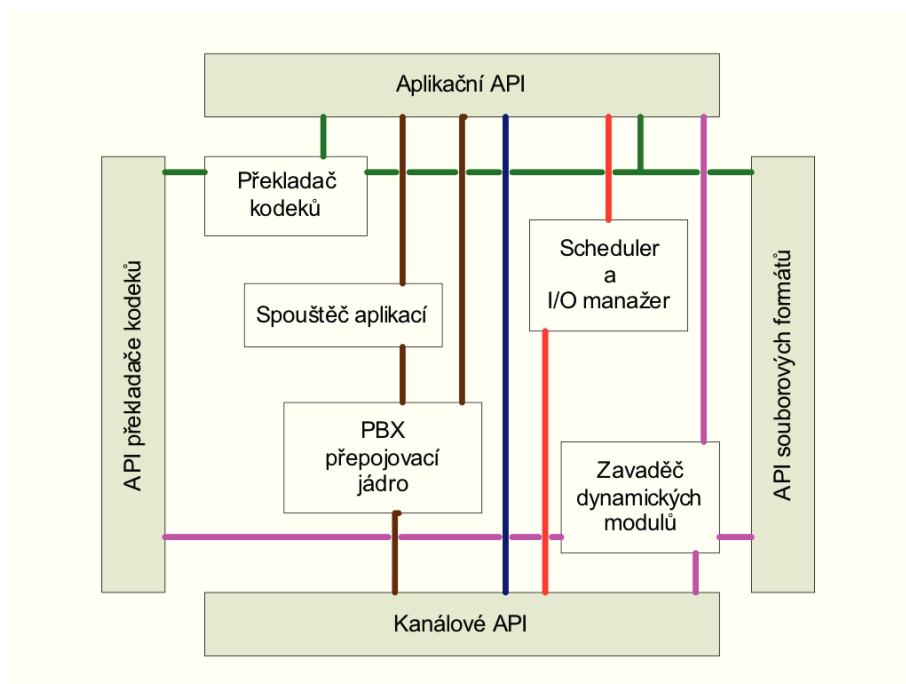
U popisu jednotlivých kodeků jsou uváděny nároky na šířku pásma, ty jsou také porovnány v tabulce 2. Tato šířka pásma je však uvažována pro kanálově propojované sítě, proto je potřeba mít na paměti, že v Ethernetu je nutno připočítat velikosti jednotlivých záhlaví spojové a síťové vrstvy. Hlas je na straně odesílatele kódován příslušným kodekem a následně paketizován do bloků o velikosti 20 až 160 bajtů. Pokud tedy chceme uvažovat šířku pásma v Ethernetu tak přičteme alespoň 66 B bez obsahu VLAN záhlaví a 68 B s použitím VLAN značkování. Pak je potřeba pro přenos v plně duplexním Ethernetu uvažovat následující šířky pásma:

- G.711 - přibližně 90 kbps
- G.723.1/ACELP kbps - přibližně 23 kbps
- G.723.1/MP-MLQ - přibližně 24 kbps
- G.729 - přibližně 35 kbps
- GSM - přibližně 40 kbps

Vezmeme-li tedy v úvahu 25 současných hovorů s kódováním PCM, tedy při použití kodeku G.711, pak minimální šířka pásma potřebná pro přenos těchto hovorů je 2,25 Mbps.

Kodek	Typ	Rychlost [kbps]	Paketizační perioda [ms]	Velikost rámce [ms]	Algoritmické zpoždění [ms]	Kodekové zpoždění [ms]
G.711	PCM	64	20	0.125	0	0.125
G.723.1	MP-MLQ	5.33	30	30	7.5	37.5
G.723.1	ACELP	6.4	30	30	7.5	37.5
G.726	ADPCM	32	20	10	0	10
G.728	LD-CELP	16	30	0.625	0	0.625
G.729A	CS-ACELP	8	20	10	5	15

Tabulka 2: Srovnání parametrů nejpoužívanějších kodeků



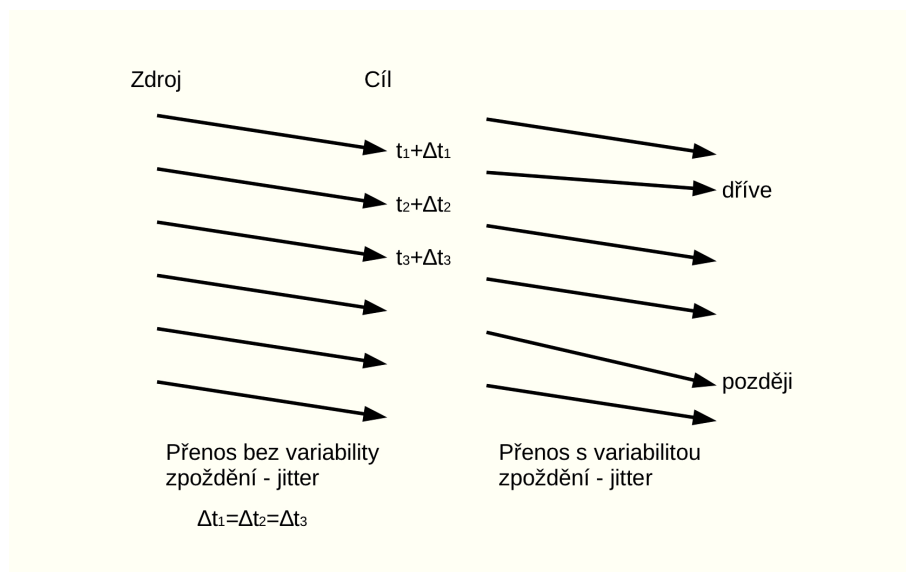
Obrázek 1: Blokové schéma architektury Asterisku (použito se souhlasem autora [3])

Pro spolehlivý přenos hlasu přes Ethernetovou síť je potřeba mít na paměti jakým způsobem Ethernet funguje, přesněji jaký typ obsluhy paketových front se nativně využívá. To je podrobněji rozebráno v podkapitole Priorizace toku dat, metody obsluhy paketových front a podpora značkování provozu v Asterisku.

2.3 Architektura Asterisku

Kolem systému centrálního jádra PBX jsou definovány specifické API jak můžeme vidět na obrázku 1. Toto jádro ovládá vnitřní propojení PBX resp. specifické protokoly, kodeky a HW rozhraní telefonních aplikací. To umožňuje v Asterisku použít libovolnou vhodnou technologii a HW (nyní i v budoucnu) za účelem vykonání základních funkcí - propojování HW a aplikací. Jádro Asterisku ovládá tyto položky [13]:

- PBX spojování (PBX Switching), hlavním cílem Asterisku je jistě propojování v PBX, spojování mezi uživateli a automatizovanými úlohami. Přepojovací jádro příslušně spojuje příchozí volání na různých HW a SW rozhraních.
- Spouštěč aplikací (Application Launcher) spouští aplikace zajišťující služby jako jsou hlasová pošta, přehrání souboru, výpis adresáře a další.
- Překladač kodeků (Codec Translator) používá moduly kodeků pro kódování a de-kódování různých zvukových kompresních formátů používaných v telefonii.



Obrázek 2: Vliv variability zpoždění - jitter

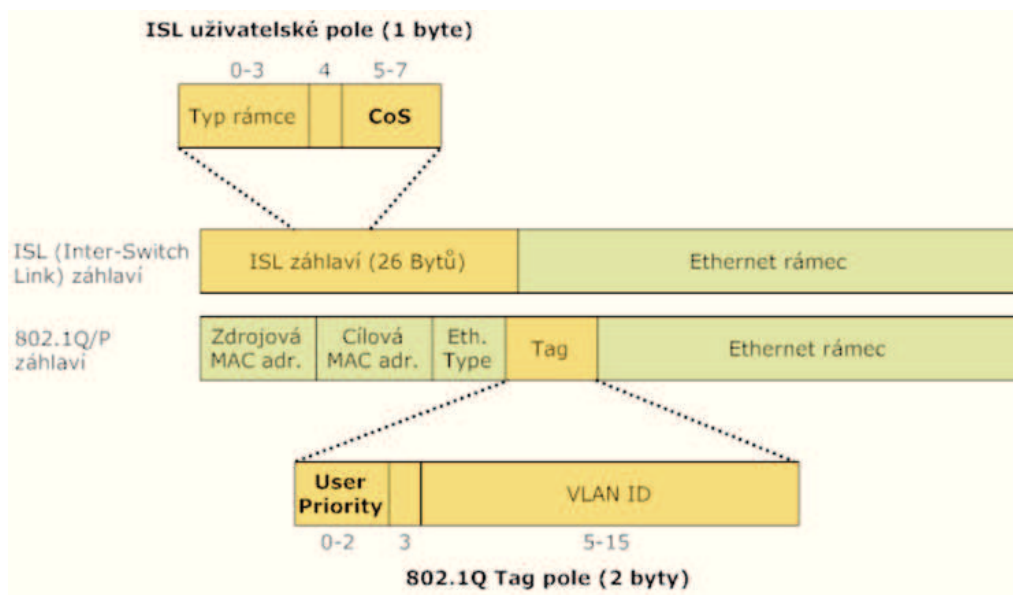
- Plánovač a I/O manažer (Schedule and I/O manager) slouží k ovládání nízkoúrovňových úloh a systémového řízení pro optimální výkon odpovídající míře zatížení.

2.4 Priorizace toku dat, metody obsluhy paketových front a podpora značkování provozu v Asterisku

Při průchodu paketu sítí vzniká několik druhů zpoždění, přes zpoždění serializační, dobu čekání v paketové frontě, kodekové zpoždění atd. Tato zpoždění se pak na přijímací straně projeví tak, že časový interval mezi přijímanými pakety z daného RTP streamu, není konstantní. Odborně je tento jev nazýván variabilita zpoždění, anglicky jitter. Pro eliminaci variability zpoždění má také Asterisk implementován tzv. de-jitter buffer, který pakety přicházející s různými časovými rozestupy ukládá do zásobníku (anglicky buffer), čímž pakety mírně zpožďuje a následně je odesílá směrem k příjemci s konstantním časovým rozestupem. Tím je dosaženo faktu, že se na přijímací straně v ideálním případě variabilita zpoždění vůbec neprojevuje. Pro lepší porozumění je variabilita zpoždění znázorněna v obrázku 2.

2.4.1 Priorizace toku dat na L2

Nyní si tedy popíšeme podporu nástroje QoS v jednotlivých technologiích na druhé vrstvě RM OSI modelu. U technologie ATM je používáno jednobitové pole CPL (Cell Loss Priority) pro prioritizaci buňek nesoucích důležitá data. Jak vyplývá z dvojkové logiky, škálovatelnost typu provozu je zde nulová, jelikož jednobitové pole může mít pouze 2



Obrázek 3: Srovnání rámců ISL a 802.1Q (použito se souhlasem autora [10])

hodnoty. Buňky které mají být prioritovány mají nastavenou hodnotu pole CPL na logickou úroveň 0 a naopak buňky, které mají nastavenou hodnotu pole CPL na 1 mohou být v případě přeplnění linky zahozeny. V sítích Frame Relay je používáno taktéž jednotového pole v záhlaví rámce Frame Relay. Toto pole se nazývá Discard Eligibility (DE) a filozofie jeho použití je shodná s použitím v sítích ATM, tedy logická 0 značí prioritní provoz, zatímco logická 1 značí provoz s nižší prioritou. Když se zaměříme na rámec technologie Ethernet je zřejmé, že v základní podobě Ethernetový rámec nepodporuje prioritaci toku na druhé vrstvě. Toto je možno pouze v případě použití enkapsulace 802.1q nebo ISL na druhé vrstvě. Přidá se tedy nové záhlaví určené převážně pro oddělení jednotlivých VLAN. V tomto záhlaví se nachází tříbitové pole Priority Code Point (PCP). To je také často nazýváno jako Class of Service (CoS) pro ISL enkapsulaci či User priority pro enkapsulaci 802.1Q. Srovnání je vidět na obrázku 3. Dle doporučení IEEE 802.1q jsou stanoveny hodnoty pole CoS/User Priority pro určité typy provozu, viz tabulka 3.

2.4.2 Priorizace toku na L3

Pro značkování provozu na třetí vrstvě RM OSI modelu je využíváno polí Type of Service, resp. Differentiated Services Code Point (ToS/DSCP). Značkování provozu dle ToS je starším standardem, který se dnes stále používá, avšak při nových implementacích je obvykle návrh realizován dle standardu Differentiated Services. Rozdíly v jednotlivých hlavičkách jsou zřejmé z obrázku 4. Zatímco u staršího modelu je k odlišení typu provozu použito tříbitové pole IP precedence a pro specifikaci požadavků na propustnost, zpoždění, spolehlivost a cenu čtyřbitové pole ToS, u novějšího modelu dle Diff-

Serv je použito šesti bitové pole DSCP kde jsou specifikovány požadavky na způsob třídění a značení paketů dle standardů Assured Forwarding (AF) a Expedited Forwarding (EF) definovaných v RFC2597 a RFC2598. Standard Assured Forwarding (AF) třídí pakety do čtyř tříd, kterým přísluší v každém směrovači čtyři paketové fronty [9].

Priorita	Zkratka	Typ provozu
1	BK	Background
0 (nejnižší)	BE	Best Effort
2	EE	Excellent Effort
3	CA	Kritické aplikace
4	VI	Video, zpoždění a variabilita < 100 ms
5	VO	Hlas, zpoždění a variabilita < 10 ms
6	IC	Internetwork Control
7 (nejvyšší)	NC	Network Control

Tabulka 3: Doporučené značkování provozu dle IEEE 802.1Q-2005

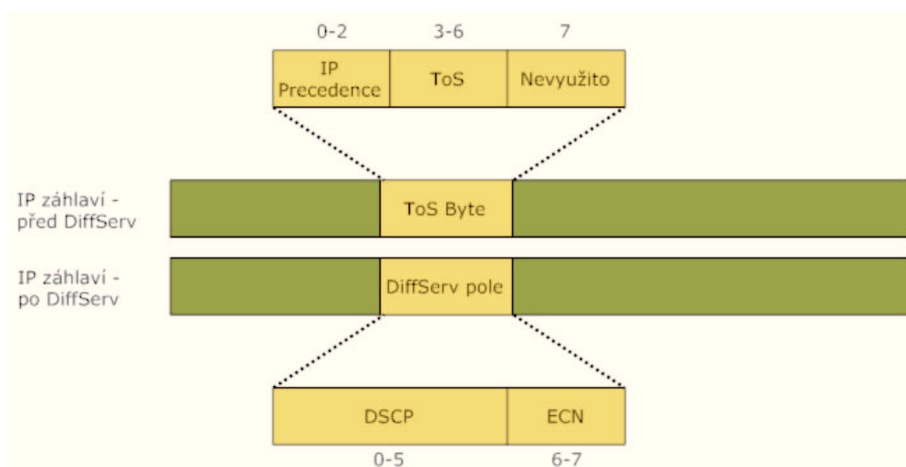
V každé z těchto tříd se se pakety dále rozdělují do tří kategorií podle pravděpodobnosti zahození mechanismem, který brání přeplnění fronty - obvykle WRED (Weighted Random Early Detection). Pro AF se tedy celkem používá dvanáct různých hodnot DSCP rozdělených jako AF xy , kde x je číslo třídy a y je číslo kategorie - čím menší číslo kategorie, tím nižší pravděpodobnost zahození. Hodnoty DSCP jsou znázorněny v tabulce 4 [10].

Pravděpodobnost zahození	Nízká	Střední	Vysoká
	Název/dek./bin.	Název/dek./bin.	Název/dek./bin.
Třída 1	AF11/10/001010	AF12/12/001100	AF13/14/001110
Třída 2	AF21/18/010010	AF22/20/010100	AF23/22/010110
Třída 3	AF31/26/011010	AF32/28/011100	AF33/30/011110
Třída 4	AF41/34/100010	AF42/36/100100	AF43/38/100110

Tabulka 4: Doporučené hodnoty DSCP pro Assured Forwarding

Třída provozu zvaná Expedited Forwarding (EF) je určena pro provoz náchylný na zpoždění, variabilitu zpoždění (jitter) a ztrátovost paketů. Tato třída využívá paketovou frontu s nejvyšší prioritou a v produkci se běžně používá výhradně pro hlasové služby. Hodnota DSCP je pro EF dekadicky 46, binárně 101110.

Další nástroj QoS popisuje starší model IntServ, jež disponuje možností rezervace přenosového pásma na dané trase. K rezervaci přenosového pásma využívá protokol RSVP (Resource Reservation Protocol), který musí podporovat všechna zařízení (většinou směrovače) na přenosové trase. V případě, že některý ze směrovačů nepodporuje protokol RSVP, může jeho zprávy jen přeposílat, případně lze v této části sítě provozovat mapování z IntServ na DiffServ. IntServ dnes není příliš používán, jelikož jeho aplikace v rozsáhlých sítích je obtížně použitelná. Spíše se můžeme setkat s kombinací obou



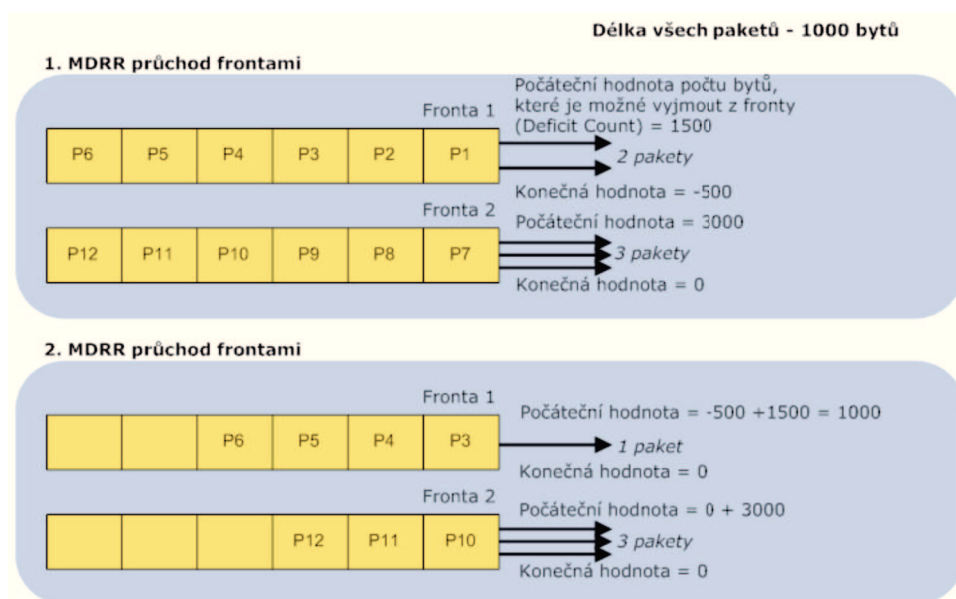
Obrázek 4: IP záhlaví s použitím a bez použití DiffServ (použito se souhlasem autora [10])

modelů, přesněji IntServ pro provoz náchylný na zpoždění - např. VoIP a DiffServ pro ostatní třídění provozu. S prioritací toku na třetí vrstvě RM OSI modelu nutně souvisí také metody obsluhy paketových front, na nichž závisí možnost prioritace toku náchylného na zpoždění.

2.4.3 Metody obsluhy paketových front

Je bezesporu, že dnes nejpoužívanější technologií v LAN (Local Area Network) sítích je Ethernet. Ethernet ve své základní podobě využívá metody pro obsluhu paketových front FIFO. FIFO je zkratkou anglických slov First-In First-Out. Jedná se tedy o klasický posuvný registr, ve kterém nikterak možno prioritovat určitý typ provozu. Pravda, využíváme-li, dnes pro většinu LAN sítí obvyklý, plně duplexní Ethernet, nemusí nás metoda obsluhy paketových front při rychlosti 100 Mbps a výše příliš trápit. Nicméně i přesto je někdy potřeba provoz v LAN sítích značkovat a tedy i změnit metodu obsluhy paketových front, především v případech použití nejrůznějších databázových systémů v LAN síti, které obvykle kladou velké nároky na šířku přenosového pásma.

Uvažujeme-li přenos hlasu, či jiného provozu náchylného na zpoždění v sítích WAN, je potřeba si uvědomit, že Ethernet je dnes ve WAN sítích stále používán minoritně. V nekorporátní sféře, tedy internet pro domácnosti, převládají bezdrátová připojení dle standardů 802.11a/b/g/n, či technologie ADSL. U korporátních uživatelů jsou to pak často sériové linky s enkapsulací Frame Relay, PPP, či HDLS, případně také SDSL/ADSL, popřípadě i bezdrátové připojení v licencovaných pásmech. Dnes nejčastěji používanou metodou obsluhy paketových front, v případě, že je aplikován nástroj QoS, je bezesporu Class-Based Weighted Fair Queuing (CBWFQ), popřípadě Low Latency Queuing (LLQ). Níže v textu si ale popíšeme všechny metody obsluhy paketových front, se kterými se v praxi můžeme setkat [11].



Obrázek 5: Princip obsluhy paketových front metodou MDRR (použito se souhlasem autora [11])

2.4.3.1 Priority Queuing(PQ)

Metoda Priority Queuing využívá čtyři fronty s různou prioritou - vysoká, střední, normální, nízká. Paket z více prioritní fronty má vždy přednost v obslužení, před pakety z méně prioritní fronty. Tím pádem hrozí možnost tzv. „vyhladovění“ front s nižší prioritou v případě, že ve frontách s vyšší prioritou bude stále provoz. Na pakety ve frontách s nižší prioritou se tedy pak nemusí dostat, nebo na ně přijde řada až ve chvíli kdy už např. nejsou nesená data potřebná (případ vypršení časového limitu TCP relace).

2.4.3.2 Custom Queuing(CQ)

Používá až šestnáct uživatelem definovaných paketových front, které jsou cyklicky obsluhovány jedna zadruhou. Z každé fronty se vybírají pakety tak dlouho, dokud není překročen limit počtu bajtů pro danou frontu, nebo dokud zbývá ve frontě nějaký paket. Pak se shodným způsobem obslouží další fronty. Tímto způsobem lze garantovat jednotlivým frontám určité procento z přenosové rychlosti daného rozhraní. V případě, že jsou některé fronty dočasně bez paketů, zbývající fronty si automaticky přerozdělí nevyužitou přenosovou pásmo. Custom Queuing však narozdíl od Priority Queuing nedokáže dostatečně upřednostnit provoz náchylný na zpoždění.

2.4.3.3 Modified Deficit Round-Robin(MDRR)

MDRR je modifikací CQ, s tím rozdílem, že řeší problém s nepřesným přerozdělením přenosové rychlosti mezi frontami v důsledku možnosti překročení limitu počtu bajtů u jednotlivých front. Bajty poslané navíc v jednom cyklu obsluhy dané fronty odečtou v

následujícím cyklu od limitu počtu vybíraných bajtů pro tuto frontu. Pro lepší pochopení můžete princip MDRR vidět na obrázku 5. Garantované procento z šířky pásma daného rozhraní pro určitou frontu je přesně dáno podílem limitu počtu bajtů pro danou frontu a součtu limitů počtu bajtů všech front. Tato metoda je obvykle využívána u páteřních gigabitových směrovačů.

2.4.3.4 Weighted Fair Queuing(WFQ)

WFQ používá pro každý datový tok, tedy data ze zdrojové IP adresy a portu aplikace na cílovou IP adresu a port aplikace, samostatnou frontu. Počet front se tedy velice rychle mění podle toho jak jednotlivé datové toky vznikají a zase zanikají. Není proto nutná klasifikace paketů. Každá fronta má přiděleno vážené procento z celkové přenosové rychlosti rozhraní, které závisí na počtu datových toků v daném okamžiku. Váha je odvozena od hodnoty v poli IP precedence. Například přenosová rychlost datového toku s IP precedence 7 je osmkrát větší než přenosová rychlost datového toku s IP precedence 0: $(7+1)/(0+1) = 8$. Popis je vysvětlen na obrázku 6.

2.4.3.5 Class-Based Weighted Fair Queuing(CBWFQ)

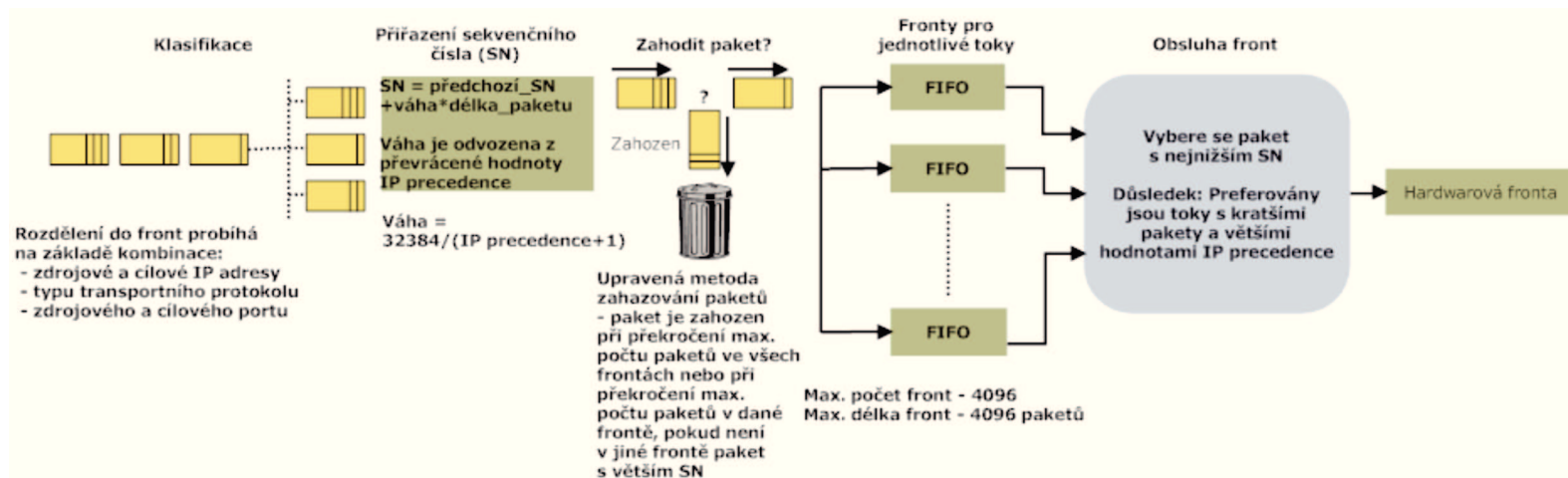
CBWFQ kombinuje výhody CQ a WFQ. Provoz je rozdělen do front, kterých může být až 64, jako u CQ avšak narozdíl od CQ je každé frontě garantováno dané procento z přenosové rychlosti rozhraní. V rámci jednotlivých front se používá buď FIFO nebo WFQ. Fronty se vyčlení pro typy provozu, kterým budeme garantovat určitou přenosovou rychlost. Tyto prioritní fronty využívají metodu FIFO. Zbytek provozu se umístí do jiné fronty, která využívá metodu WFQ. Nedostatkem zůstává stále nedostačující možnost upřednostnění provozu citlivého na zpoždění. CBWFQ je však v tomto ohledu lepší než dříve uvedené metody s výjimkou PQ. CBWFQ je často používanou metodou a využívá se například u CE, pro optimální prioritizaci toku dat.

2.4.3.6 Low Latency Queuing(LLQ)

LLQ jen doplňuje metodu CBWFQ o frontu, popř. více front, ve které mají data náchylná na zpoždění absolutní přednost před ostatním provozem. Tato fronta se označuje jako LLQ fronta. Funkce fronty LLQ je téměř shodná s funkcí fronty s vysokou prioritou u metody PQ. Aby nedošlo k tzv. „vyhladovění“ ostatních front, jak se tomu může stát u PQ, definuje se pro LLQ maximální přenosová rychlost po jejímž překročení se přebývající pakety zahodí. Proto je třeba tuto rychlost vhodně stanovit s ohledem potřeby uživatelů sítě. Vhodné je také použití QoS nástroje pro řízení přijetí hovorů (Call Admission Control). Metoda obsluhy paketových front LLQ je nejčastěji používanou metodou u těch CE směrovačů v sítích poskytovatelů MPLS VPN, kde je potřeba dostatečně upřednostnit hlasový provoz.

2.4.4 Podpora značkování provozu v Asterisk

Asterisk podporuje oba druhy klasifikace provozu, tedy na druhé i na třetí vrstvě RM OSI modelu. Podpora značkování provozu je v Asterisku dostupná jak pro signalizaci



Obrázek 6: Princip obsluhy paketových front metodou WFQ (použito se souhlasem autora [11])

tak pro samotný hovor (RTP stream). U všech paketů odcházejících z Asterisk PBX může být nastaveno pole ToS v záhlaví IP paketu. Protože je Asterisk nativně provozován na Linuxu, může také značkovat odchozí provoz na druhé vrstvě, dle 802.1p, tedy pole CoS ve VLAN záhlaví rámce. Toto je užitečné především v korporátní sféře, kdy se využívá prioritizace VoIP hovorů v rámci dané LAN před ostatním provozem. Asterisk však nastaví pouze prioritu pro Linuxový soket, který je potřeba dále mapovat na samotné VLAN záhlaví. Toto mapování je možné provést příkazem ve výpisu 1 [12]:

```
root@HomeRouter# vconfig set.egress_map [vlan-device] [skb-priority] [vlan-qos]
```

Výpis 1: Příkaz pro mapování Linux socket priority na 802.1p CoS

Následující tabulky 5 a 6 ukazují všechny kanálové ovladače a přídatné moduly Asterisku, které podporují nastavení QoS pro odchozí provoz. Také je zobrazen typ provozu, který je podporován jednotlivými moduly.

	Signalizace	Audio	Video	Text
chan_sip	Ano	Ano	Ano	Ano
chan_skinny	Ano	Ano	Ano	Ne
chan_mgcp	Ano	Ano	Ne	Ne
chan_unistm	Ano	Ano	Ne	Ne
chan_h323	Ne	Ano	Ne	Ne
chan_iax2	Ano	Ne	Ne	Ne

Tabulka 5: Kanálové ovladače podporující QoS značkování

	Signalizace	Audio	Video	Text
dundi.conf	Ano	Ne	Ne	Ne
iaxprov.conf	Ano	Ne	Ne	Ne

Tabulka 6: Konfigurační soubory podporující nastavení pole ToS pro signalizaci

Podporované hodnoty pole **ToS** jsou: CS0, CS1, CS2, CS3, CS4, CS5, CS6, CS7, AF11, AF12, AF13, AF21, AF22, AF23, AF31, AF32, AF33, AF41, AF42, AF43 a EF (expedited forwarding). ToS parametry je možno zadat také v číselné podobě - např. EF=184, AF41=136 atd.

Protože 802.1p používá 3 bity z VLAN hlavičky, může toto pole nabývat hodnot od 0 do 7. Doporučené hodnoty pro značkování provozu přímo v Asterisku jsou uvedeny v tabulce 7.

	ToS	CoS
Signalizace	cs3	3
Audio	ef	5
Video	af41	4
Text	af41	3

Tabulka 7: Doporučené hodnoty ToS a CoS pro odchozí provoz z Asterisku

Poznámka 2.1 Pokud chceme v Asterisku využívat ToS značkování provozu, musí být Asterisk spuštěn pod uživatelem **root**, nebo musí být zkompileován s knihovnou *libcap*

2.4.4.1 Povolení značkování provozu v konfiguračních souborech Asterisku

Pro aktivaci značkování signalizace protokolu IAX je nutno v konfiguračním souboru *iax.conf* povolit *tos* parametr a přiřadit mu hodnotu. Tímto tedy globálně aktivujeme doporučenou hodnotu značkování (viz tabulka 7) pro IAX pakety generované kanálem *chan_iax2*. Jelikož IAX spojení kombinují signalizaci, audio i video do jednoho UDP streamu, není možné nastavit hodnotu ToS pro každý typ provozu samostatně. Pro globální povolení značkování provozu v konfiguračním souboru *iax.conf* je potřeba odkomentovat řádky ToS pro třetí vrstvu RM OSI modelu a CoS pro druhou vrstvu zmiňovaného modelu - viz výpis 2.

```

; See qos.tex or Quality of Service section of asterisk.pdf for a description of these
; parameters.
tos=ef
cos=5

```

Výpis 2: Parametry v konfiguračním souboru *iax.conf* pro nastavení ToS a CoS

V konfiguračním souboru *sip.conf* jsou dostupné čtyři parametry pro značkování provozu a to signalizace, audio, videa i textu. Parametr *tos_sip* nastavuje ToS hodnotu pro signalizaci. Parametry *tos_audio*, *tos_video* a *tos_text* pak nastavují ToS hodnoty pro RTP pakety nesoucí audio, video, či text. V souboru *sip.conf* jsou také dostupné čtyři parametry pro značkování na druhé vrstvě RM OSI modelu, dle 802.1p CoS: *cos_sip*, *cos_audio*, *cos_video* and *cos_text*. Pro tyto parametry platí stejná pravidla jako pro parametry nastavující ToS. Ve výpisu 3 jsou uvedeny příklady konfigurace ToS a CoS.

```

tos_sip=cs3      ; Sets TOS for SIP packets.
tos_audio=ef     ; Sets TOS for RTP audio packets.
tos_video=af41   ; Sets TOS for RTP video packets.
cos_sip=3        ; Sets 802.1p priority for SIP packets.
cos_audio=5      ; Sets 802.1p priority for RTP audio packets.
cos_video=4      ; Sets 802.1p priority for RTP video packets.
cos_text=3       ; Sets 802.1p priority for RTP text packets.

```

Výpis 3: Parametry v konfiguračním souboru *sip.conf* pro nastavení ToS a CoS

Jak je patrné z výše uvedených konfigurací a příkazů, Asterisk podporuje všechny potřebné nástroje pro značkování provozu, které k bezchybně funkční VoIP infrastruktuře neodmyslitelně patří. Stává se proto vhodným a mocným nástrojem pro firemní telefonii i v těch nejrozsáhlejších korporátních sítích.

3 HW platforma ALIX a green computing

Hardwarová platforma nesoucí název ALIX, dříve WRAP, je vyvíjena a distribuována nyní švýcarskou firmou PC Engines. Firma PC Engines byla založena roku 1995 Pascalem Dornierem ve Spojených státech Amerických. Roku 2002 potom přenesla svou výrobu do Švýcarska pod novým jménem PC Engines GmbH, pod kterým na trhu vystupuje dodnes. Firma PC Engines se dnes zaměřuje výhradně na výrobu Embedded systému, konkrétně na výrobu komplexních minimalizovaných základních desek pracujících na architektuře x86 a nesoucích název ALIX (dříve WRAP). Její zakladatel Pascal Dornier je držitelem několika desítek patentů, mnohdy souvisejících právě s myšlenkou co nejefektivnějšího využívání elektrické energie v počítačových systémech [17]. Dnes se tato větev vývoje Embedded systémů nazývá **Green Computing**. Desky ALIX využívají vlastního BIOSu pro zavádění systému a POST testy. Tento BIOS nese název tinyBIOS, je vyvíjen firmou PC Engines a je distribuován pod licencí CPL (Common Public License). Licence CPL je obecnou verzí licence IBM Public License. Jak výrobce sám v dokumentu o tinyBIOSu prezentuje - tinyBIOS je určen pro použití s open-source operačními systémy, nikoli Microsoft Windows [16]:

„tinyBIOS was written from the ground up to meet the requirements of embedded PC systems, based on my experience in the PC industry and with embedded PC applications. tinyBIOS is designed for use with open source operating systems, not Microsoft Windows.“

Z výše uvedené charakteristiky tedy vyplývá, že nejen projekt tinyBIOS, ale produkty firmy PC Engines obecně jsou určeny především pro síťové, případně miniserverové systémy spíše, než pro desktopy - ačkoli existuje produktová řada desek ALIX určených pro desktopové prostředí, viz podkapitola Platforma ALIX.

3.1 Platforma ALIX

Firma PC Engines v současnosti nabízí širokou škálu desek ALIX, které se více či méně liší svými hardwarovými komponenty. Společným rysem všech nabízených modelů je procesor AMD Geode LX800 a tinyBIOS (s výjimkou dvou "desktopových" modelů, které využívají Award BIOS). Dalším společným parametrem dnes nabízených desek je velikost operační paměti 256 MB, což je pro aplikaci v prostředí počítačových sítí rozhodně dostatečná velikost. Přehled vyráběných modelů je v tabulce 8 [17].

Model ALIX	CPU	DRAM	LAN	mini PCI	PCI	USB	jiné	BIOS	rozměr
1d	LX800	256MB	1	1	1	2	VGA, audio, PS/2, LPT, GPIO	Award	mini ITX
2d2	LX800	256MB	2	2	0	2		tiny	6x6"
2d3	LX800	256MB	3	1	0	2		tiny	6x6"
2d13	LX800	256MB	3	1	0	2	baterie, I2C, COM2, interní USB	tiny	6x6"
3d2	LX800	256MB	1	2	0	2		tiny	100 x 160 mm
3d3	LX800	256MB	1	2	0	2	VGA, audio	Award	100 x 160 mm
6f2	LX800	256MB	2	1	0	2	miniPCI express	tiny	6x6"

Tabulka 8: Modely desek ALIX

Z tabulky 8 je patrné, že dva typy - konkrétně ALIX.1D a ALIX.3D3 disponují namísto tinyBIOSu Award BIOSem. Důvod je jednoduchý, jelikož právě tyto dva systémy jsou navrženy pro desktopové prostředí, tedy možnost aplikovat jiný než open-source operační systém (např. i Microsoft Windows). Pro tuto diplomovou práci byl zvolen model ALIX.2D13 zobrazen na obrázku 10.

Poznámka 3.1 Osobně jsem měl možnost ALIX.1D vyzkoušet. Nainstalovaným systémem byla linuxová distribuce Lubuntu ve verzi 10.04 která využívá grafického prostředí LXDE. Systém není příliš svižný, jelikož 256 MB RAM sdílené s grafickou kartou VGA je pro desktopový počítač přeci jen spíše ohlasem dob minulých, nicméně jako kioskové PC, či tenký klient je tato platforma použitelná. Výrobce podporovanou kompatibilitu s MS Windows XP Home jsem bohužel neměl možnost odzkoušet. Avšak na stránkách výrobce procesoru - firmy AMD, je dostupný benchmarkový test - viz podkapitola AMD Geode LX 800, obrázek 9.

3.1.1 tinyBIOS

Jak již bylo naznačeno, tinyBIOS je určen speciálně pro embedded systémy, užívané především pro síťové aplikace, a to nejen od výrobce PC Engines. Nicméně tinyBIOS je společným „mozkem“ všech desek ALIX. Základní vlastnosti a výhody tinyBIOSu jsou tyto [18]:

- Neobsluhované funkce - žádné chyby typu: *„keyboard failure - press F1 to continue“*
- Možnost přístupu do zdrojového kódu znamená snadnou adaptaci a odstraňování chyb při implementaci na daný hardware
- Absence konfigurační obrazovky eliminuje náklady na nastavení a selhání CMOS RAM
- menší velikost (16 až 32 KB) znamená více volného místa pro aplikace (OS)
- menší zdrojový kód (okolo 10000 řádků) znamená snažší porozumění a orientaci v kódu
- Použití A386 překladače k překladu celého BIOSu v jedné vteřině

Jádro tinyBIOSu je distribuováno pod licencí CPL jak již bylo zmíněno výše. Podpora pro první dva používané chipsety je zdarma, další jsou pak zpoplatněny viz tabulka 9. Moduly jednotlivých čipsetů pro ostatní části mohou být doimplementovány uživatelem a případně i licencovány firmou PC Engines. Aktuálně podporovány tyto čipsety jsou v tabulce 9.

Výrobce	Čipset	Cena
Acer Labs	ALI M1487 FINALi ALI M6117 ALI Aladdin V	Zdarma, open source Zdarma, open source Na dotaz - nedoporučeno pro nové verze
AMD	Elan SC400 / SC410 Elan SC520 GXM SC1100 SC1200	Na dotaz - nedoporučeno pro nové verze \$1995 Na dotaz - vyžadovány nativní ovladače pro video a audio \$1995 - vyžadován nativní audio ovladač \$1995 - vyžadován nativní audio ovladač
Intel	430TX	\$1995
ST Microelectronics	STPC Consumer II	\$1995
	STPC Elite/Atlas	zahrnuto v STPC Consumer II
	STPC Client, Consumer	nedoporučeno pro nové verze
	STPC Vega	\$1995
VIA Technologies	CLE266 / C3 CPU	\$2995 - dostupný předběžný testovací kód
ZF Micro Solutions	ZFx86	Na dotaz - nedoporučeno pro nové verze

Tabulka 9: Čipsety podporované pro použití s tinyBIOS

3.1.2 AMD Geode LX800

Rodina procesorů AMD Geode je sérií x86-kompatibilních tzv. „system-on-a-chip“ mikroprocesorů vyvíjena společností AMD a zaměřena na Embedded počítačové systémy. Vývoj této série byl původně započat firmou National Semiconductor v roce 1999. Samotné jádro procesorů Geode je odvozeno od platformy Cyrix MediaGX která byla získána celostátním sloučením s firmou Cyrix v roce 1997. AMD rozděluje rodinu procesorů Geode do dvou tříd:

- MediaGX-odvozené: Geode GX a LX modely
- Athlon-odvozené: Geode NX modely

Geode procesory jsou optimalizované pro minimální spotřebu elektrické energie a tedy nízké provozní náklady, přičemž ale stále zůstávají kompatibilní se software, který byl napsán pro platformu x86. Procesory Geode GX a LX, tedy procesory odvozené z MediaGX, postrádají moderní vlastnosti procesorů jako je SSE (Streaming SIMD Extensions) či větší integrovaná L1 cache. Právě proto je zde novější nabízená řada Geode



Obrázek 7: AMD Geode LX 800 (500MHz) CPU

NX, odvozena od jádra procesorů Athlon, která všechny dnes běžné vlastnosti procesorů nabízí. Rodina procesorů Geode je tedy nejlépe vhodná pro tenké klienty, set-top boxy a ostatní embedded systémy jako je například neobvyklý projekt Nao robot, či Win Enterprise IP-PBX [19]. Pouzdro procesorů AMD Geode LX můžeme vidět na obrázku 7.

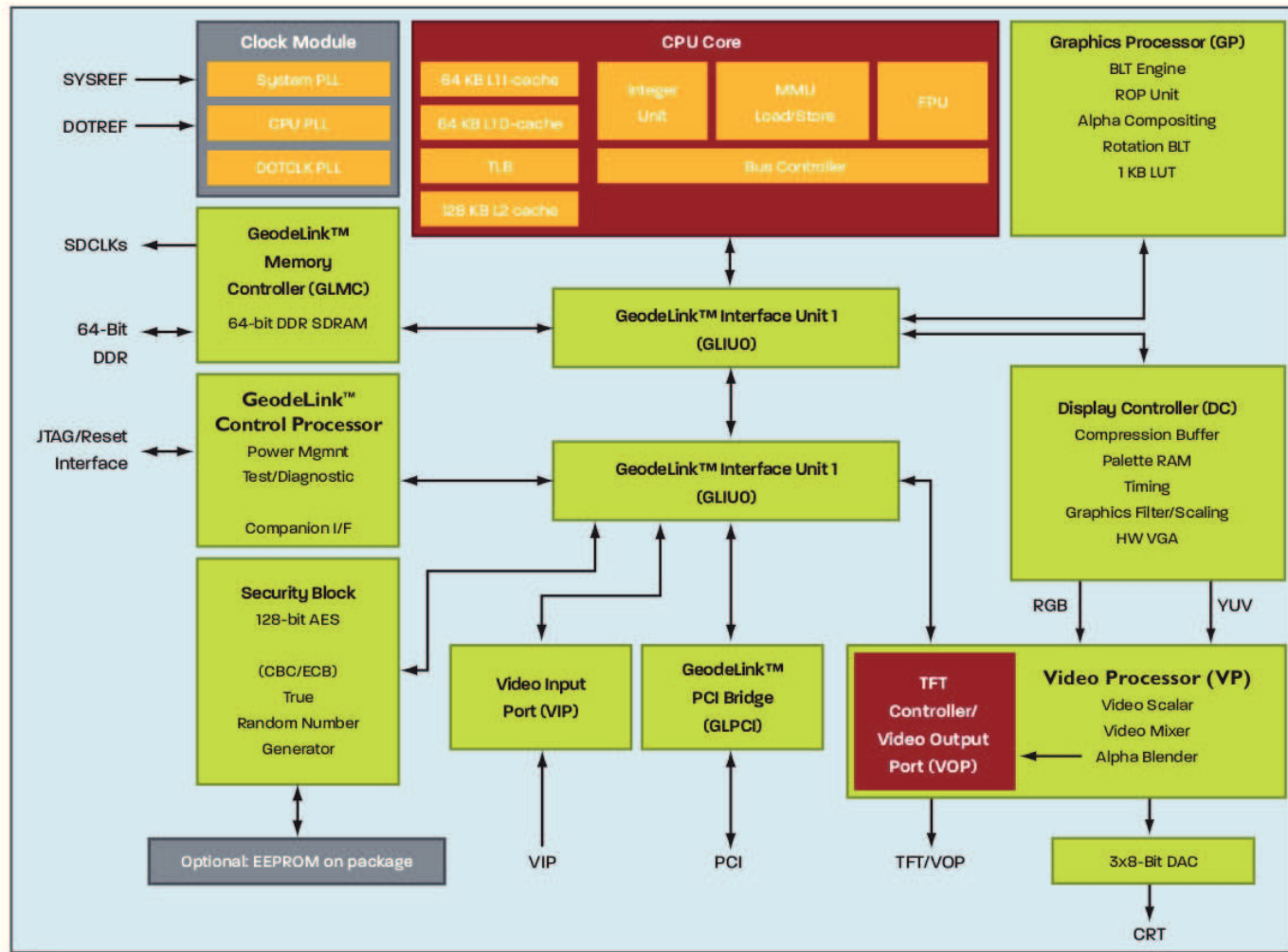
Procesor **AMD Geode LX800** jenž je použit ve všech deskách ALIX, disponuje následujícími vlastnostmi [20]:

- Nízká spotřeba elektrické energie - 1.8 wattů (TDP 3.6 W)
- Plná x86/x87 kompatibilita
- Funkční bloky procesorové jednotky:
 - CPU jádro
 - GeodeLink řídicí procesor
 - GeodeLink jednotka rozhraní
 - GeodeLink paměťový kontroler
 - Grafický procesor
 - Kontroler displeje
 - Video procesor
 - Video vstupní port
 - GeodeLink PCI můstek
- Bezpečnostní blok:
 - 128-Bit Advanced Encryption Standard (AES) - (CBC/ECB)
 - Generátor náhodných čísel (True Random Number Generato)

Specifikace procesoru AMD Geode LX800:

- Frekvence procesoru 500 MHz
- Power management: ACPI, nižší spotřeba, funkce wakeup SMI/INTR.
- 64KB pro instrukce / 64KB pro data L1 cache a 128KB L2 cache
- Oddělený instrukční a datový prostor - cache/TLB.
- 64-bitová sběrnice DDR paměti - 400 MHz
- Integrovaný FPU s MMX a 3DNow!
- 9 GB/s interní GeodeLink Interface Unit (GLIU)
- CRT a TFT (vysoké a standardní rozlišení) - Podpora 1920x1440 v CRT módu a 1600x1200 v TFT módu
- VESA 1.1 a 2.0 VIP/VDA podpora
- Vyroben technologií 0,13 mikronů
- 481-terminal PBGA (Plastic Ball grid array)
- GeodeLink active hardware power management
- kompatibilní se základními deskami s patičí Socket 7

Jak je uvedeno v popisu výše a zobrazeno na obrázku 8, architektura procesorů AMD Geode LX disponuje video procesorem zvládajícím zobrazení vysokého rozlišení až 1920x1440 v módu CRT. Toto je užitečné pochopitelně pouze pro desktopové aplikace, či aplikaci v průmyslových aplikacích. Pro námi požadované síťové aplikace je spíše důležité se zaměřit na jednotku Security Block zobrazenou v levé spodní části obrázku 8. Tato může mít hojně využití především pro síťové služby typu VPN využívající AES šifrování. Podporovány jsou oba používané způsoby AES šifrování a to jak ECB (Electronic Code Book) tak CBC (Cipher Block Chaining). Při použití ECB se posloupnost bloků otevřeného textu šifruje tak, že každý blok je šifrován zvlášť. Nevýhodou takového typu šifrování je, že stejné bloky otevřeného textu mají vždy stejný šifrový obraz. Zatímco u typu šifrování CBC je nejprve každý blok otevřeného textu modifikován předchozím blokem šifrového textu pomocí modusu, a teprve poté se šifruje. Budeme-li šifrovat jeden a tentýž, byť i velmi dlouhý, otevřený soubor dat dvakrát, obdržíme naprosto odlišný šifrový text. CBC je v současné době nejpoužívanějším operačním modem blokových šifer [22]. Otevřeným open-source řešením virtuální privátní sítě, které využívá šifrování AES je pro linuxové distribuce velmi oblíbená OpenVPN. Instalace OpenVPN je ve všech běžně dostupných distribucích linuxu prováděna přes repozitář příslušné distribuce, nebo je ji samozřejmě možno zkompilovat. Více o implementaci OpenVPN na minimalizovaný systém HomeRouter je popsáno v poslední kapitole Možnosti rozšíření Embedded Asterisk PBX. Dalším



Obrázek 8: Blokové schéma architektury AMD Geode LX [21]

otevřeným řešením pro implementaci VPN užívající AES šifrování je například projekt OpenSWAN, který je opět dostupný z repozitáře pro většinu běžně používaných linuxových distribucí [22].

Samotný výpočetní výkon rodiny procesorů AMD Geode LX je pochopitelně značně omezený s ohledem na jeho minimální energetické nároky. Výrobce na svých stránkách uvádí benchmarkový test, který můžete vidět na obrázku 9. V tomto testu byl srovnáván procesor AMD Geode LX800 používán v základních deskách ALIX s procesorem AMD Geode GX533 [21].

Poznámka 3.2 Nutno poznamenat, že procesor AMD Geode GX533 pracuje na frekvenci 400 MHz, zatímco procesor AMD Geode LX800 na frekvenci 500 MHz.

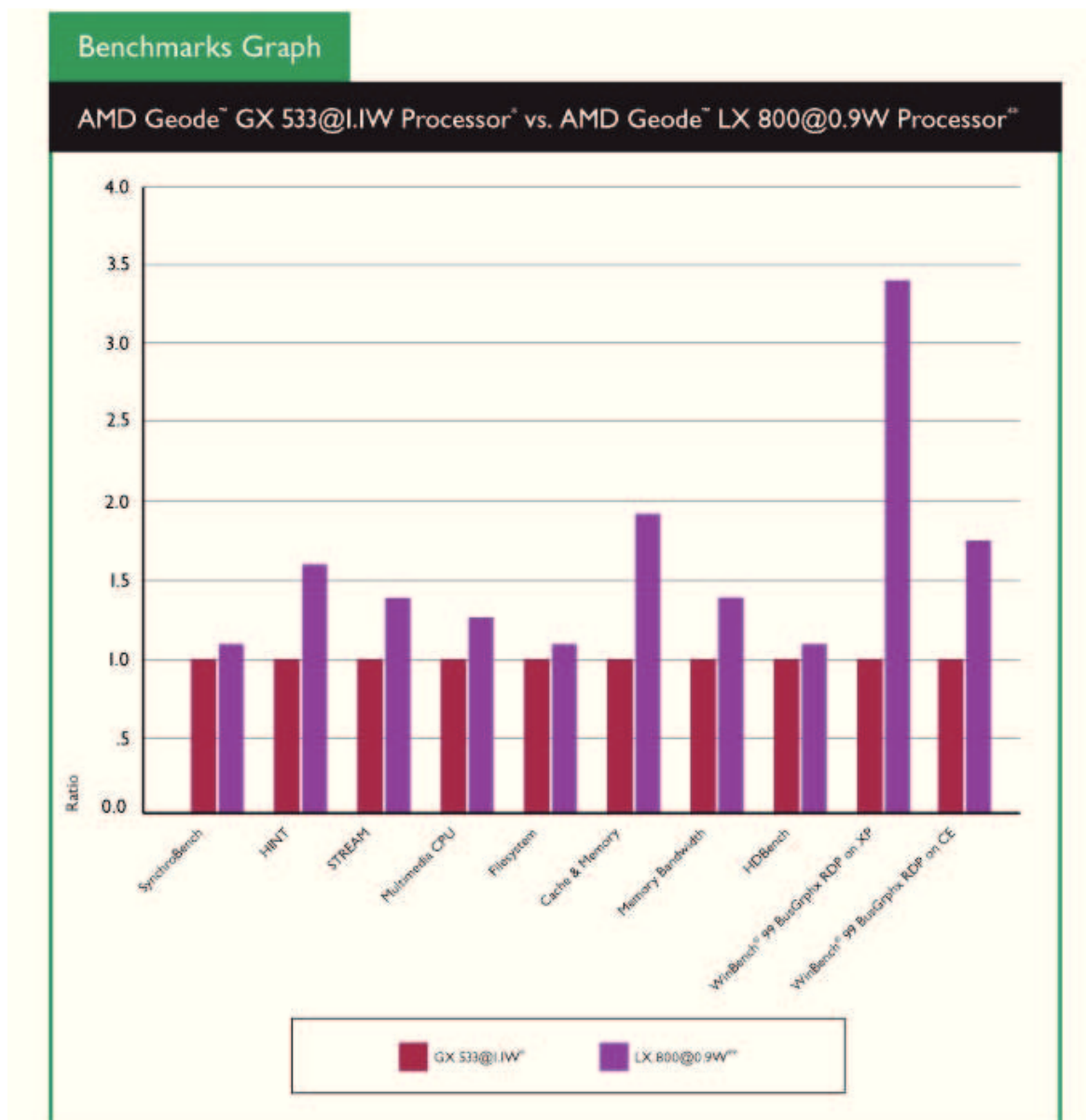
Oba systémy byly testovány na systémech Microsoft Windows XP Home a Windows CE, při použití nástroje WinBench [23]. Konfigurace jednotlivých systémů:

- **AMD Geode GX533@1.1W procesor** - Frekv.:400 MHz, RAM: DDR 256 MB (sdílená), HDD: Western Digital 80 GB, Zákl.deska: Hawk, OS: Microsoft Windows XP Home, Windows CE
- **AMD Geode LX800@0.9W procesor** - Frekv.:500 MHz, RAM: DDR 256 MB (sdílená), HDD: Western Digital 80 GB, Zákl.deska: Hawk, OS: Microsoft Windows XP Home, Windows CE

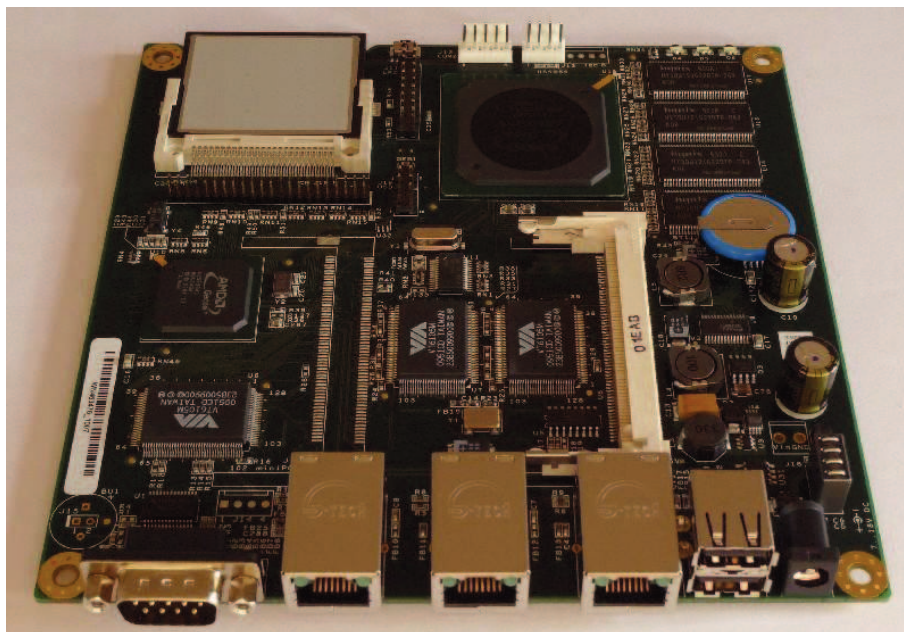
Dosažené výsledky v jednotlivých oblastech výkonu jsou zobrazeny sloupcovým grafem na obrázku 9. Nejlepších výsledků dosahuje AMD Geode LX v testu Winbench®'99 BusGrphx RDP na OS Windows XP. Jedná se o test, který je poslán v binárním formátu přes Ethernetovou síť do testovaného objektu a následně vyhodnocen na serveru z něhož byl test zadán. Jednotlivé testy prováděné v tomto srovnání jsou podrobněji popsány v dokumentu [24].

3.2 Green Computing

Green Computing nebo také Green ICT (Information and Communication Technologies) je nejdůležitějším inovačním fenoménem ICT v poslední době. Orientaci na tzv. green ICT můžeme sledovat již dlouhodoběji a jejich podpora se objevuje výrazněji v příspěvcích a konferencích již od roku 2008, resp. 2009. Jejich důležitost však zdůraznila stále hrozící ekonomická krize a s ní související hledání všech vhodných možností řešení s dlouhodobým dopadem, např. ve snaze snížit náklady na spotřebovanou energii. V souvislosti s diskutovanou možností energetické krize je „green growth“ jednou z důležitých možností jak oddělit ekonomický růst od spotřeby energie včetně dalšího vlivu na prostředí. Mnoho států si v období ekonomické krize uvědomilo, že pro jejich další růst je potřebné hledat odlišné zdroje a přístupy k vymanění se ze závislosti na elektrické energii z vyčerpatelných zdrojů. Jedním z těchto přístupů jsou investice ze strany jednotlivých států, např. do obnovitelných zdrojů energie [25].



Obrázek 9: Test výkonosti AMD Geode LX800 uváděn výrobcem [21]



Obrázek 10: ALIX.2D13

Green computing neboli green ICT se vztahují k environmentálně udržitelnému rozvoji a zahrnují efektivní návrh, výrobu a užití počítačů, serverů a ostatních ICT zařízení jako jsou monitory, tiskárny, zařízení pro uložení dat, síťové a komunikační systémy tak, aby se minimalizoval vliv na okolí a na životní prostředí. Celkově jsou řešení green ICT komplexní a zahrnují interdisciplinární přístup. Příkladem může být:

- zlepšení účinnosti ICT infrastruktury na základě využití energeticky účinných zdrojů a zařízení, alokováním vhodného množství zdrojů pro zajištění požadovaného pracovního výkonu (např. automatické optimalizace výkonu)
- vývoj nových technologií jako jsou nové materiály, nízko napěťová zařízení, zvýšení integrace (např. umístění celého systému na jednom čipu) a power aware ICT (a to jak v oblasti hardware, tak i software)
- zlepšení chování koncových uživatelů s ohledem na lepší využívání energií a snižování její spotřeby, kde ICT mohou pomoci při monitoringu a také např. při vypínání nevyužívaných zařízení
- zvýšení sociálního uvědomění pomocí komunikace a přístupu objasňujícího naléhavost tohoto tématu

ICT mají schopnost redukovat vliv na okolní prostředí za předpokladu změny chování podniků i změny životního stylu, pracovních a životních návyků a zvyků obyvatelstva.

Není bez zajímavosti, že oblast ICT je sama velkým odběratelem energie s výrazným podílem na celkové globální spotřebě. Pozornost si v tomto směru zaslouhuje i samotný internet. Z hlediska spotřeby se udává množství energie v hodnotě 153 TWh nutné pro chod jeho serverů a datových center. To představuje 1 % globální spotřeby elektrické energie. K tomu je nutno zohlednit, že provozovatelé těchto serverů a datových center v letech 2000 až 2005 svoji spotřebu elektřiny zdvojnásobili a tento trend nárůstu pokračuje i nyní. Ačkoli i virtualizace serverů zažívá období rozkvětu, což je v tomto ohledu jistě přínosem. Jen v rámci EU je spotřeba výše zmíněných zařízení odhadována na 1,4 % celkové spotřeby elektřiny ve výši 39 TWh.

Dále je potřeba si uvědomit, že vedle datových a webových center připadá největší spotřeba energie na domácí zařízení - TV přístroje, osobní počítače, mobilní telefony, Wi-fi AP a směrovače, stejně tak jako na dnes stále populárnější domácí NAS servery (počet všech těchto zařízení se odhaduje celosvětově na čtyři miliardy). ICT produkty a služby spotřebovávají necelých 8 % elektřiny v EU a odhaduje se, že by do roku 2020 mohla spotřeba vzrůst na 10,5 % .

V tomto ohledu pak technologie ICT mají nemalý potenciál i pro vlastní úspory. Dnešní zařízení jsou vysoce náročná na provoz. Připomeňme, že pár desítek let zpátky musely například aplikační programy daleko více šetřit paměť počítače i výkon jeho procesoru, které v té době byly limitní. Další oblastí úspor může být předimenzovaná ICT infrastruktura, která bývá v současnosti v podnicích zdvojena až ztrojena. V důsledku toho je následně využívána jen z poloviny, resp. třetiny. V neposlední řadě dalším uváděným potenciálem úspor je nevyužití vlastního, relativně vysokého, výkonu počítačů – odhady se pohybují na úrovni 20 % [25].

S ohledem na všechny výše zmíněné aspekty je potřeba uvítat a podporovat jakákoliv nová řešení zabývající se problematikou Green Computing. Proto je i tato diplomová práce soustředěna na minimalizaci provozních nákladů spojených s provozem Asterisk PBX. A co víc, tato práce sahá i trochu dále a to k integraci několika síťových služeb do jednoho miniserveru s názvem HomeRouter, který je zaměřen do SOHO segmentu trhu. Tato rozšíření o další síťové služby jsou dále popsána v sekci Možnosti rozšíření Embedded Asterisk PBX.

4 Implementace Asterisku na HW ALIX

Pro tuto diplomovou práci a projekt s názvem HomeRouter byla zvolena produktová řada ALIX.2D13 (na obrázku 10) společně s operačním systémem Voyage Linux. Tato základní deska disponuje následujícím vybavením:

- CPU: 500 MHz AMD Geode LX800
- RAM: 256 MB DDR DRAM
- HDD: CompactFlash socket nebo 44 pin IDE header
- Napájení: DC jack nebo pasivní POE, min. 7V max. 20V
- Tři přední LED diody, resetovací tlačítko
- 1 miniPCI slot, LPC sběrnice, I2C sběrnice, piezo, RTC baterie
- 3 Ethernetová rozhraní (Via VT6105M 10/100)
- I/O: DB9 sériový port, duální USB port (2 USB konektory vyvedeny v zadním panelu, 2 dostupné na desce pomocí pinů)
- Rozměr: 6 x 6" (152.4 x 152.4 mm)
- Firmware: tinyBIOS

V případě HomeRouter je OS Voyage linux uložen na CF(Compact Flash) kartě o velikosti 8 GB. Po základní instalaci systému a aktualizací, společně s nainstalovaným Asteriskem a jeho GUI dostupným pro Voyage linux, je využito přibližně 315 MB z celkové kapacity CF karty. Po nainstalování všech ostatních rozšíření, popsanych v poslední kapitole Možnosti rozšíření Embedded Asterisk PBX, zabírá systém na disku cca. 1,7 GB.

4.1 Voyage Linux

Voyage Linux je postaven, podobně jako distribuce Ubuntu, na spolehlivé a léty ověřené platformě linuxu s názvem Debian. Voyage Linux je minimalizovaným Debian linuxem, který po instalaci obsahuje pouze základní balíky a moduly nutné pro samotný chod systému. Velikost této základní instalace by neměla překročit 128 MB. Jak uvádí distributor Voyage Linux na svých stránkách [26]:

„Voyage is a very stripped-down Debian Linux ... With Voyage you have the entire world of Debian available to you, so customizing your own gear is easy. It's great for firewalls and routers, and specialized servers that need a small footprint.“

Carla Schroder, author of Linux Networking Cookbook



Obrázek 11: Logo Voyage Linux

Voyage Linux je určen k provozování na x86 embedded platformách jako jsou PC Engines ALIX/WRAP, Soekris 45xx/48xx a systémy postavené na jádrech procesorů Intel Atom. Aplikací na tyto embedded systémy se Voyage Linux také otevřeně hlásí k myšlence Green Computing, jak na svých stránkách ostatně také uvádí. Logo Voyage Linuxu je na obrázku 11.

Voyage Linux je plně modulární a umožňuje libovolnou rozšiřitelnost o jakýkoli balík určen pro distribuci Debian, popřípadě Ubuntu. Samozřejmě lze do této verze OS zkompileovat libovlný software určen pro linuxové systémy. V současné době Voyage Linux nabízí tyto tři edice:

- **Voyage Linux** - základní verze systému
- **Voyage MPD** - Music Player Daemon (speciální verze obsahující poslední ovladače ALSA a podporu USB Audio Class 2 pro USB zvukové karty)
- **Voyage ONE** - Verze určená pro VoIP ústřednu - Asterisk včetně GUI dostupný okamžitě po instalaci

Všechny tyto distribuce jsou dostupné buď jako *.tar* komprimované, nebo jako LiveCD pro i386 architektury. AMD64 architektura je dostupná pouze ve verzi Voyage Linux. Distributor také nabízí SDK (Software Development Kit) pro snadnější přizpůsobení Voyage Linuxu na vybranou platformu. Aktuálně poslední verze je 0.8.0 využívající kernel 3.0.0.

4.2 Instalace Voyage Linux na ALIX.2D13

Jak již bylo uvedeno v popisu desky ALIX.2D13, tento typ nedisponuje VGA rozhraním. Proto lze pro instalaci Voyage Linux na ALIX.2D13 využít aktuálně tři způsobů:

- **Instalace ze zaváděcího USB paměťového média s obsahem LiveCD** přímo v ALIX.2D13, ovládání přes terminál připojený skrz RS232 (38400 8N1, kontrola toku dat vypnuta)
- **PXE boot** – stisk E v menu tinyBIOSu, nebo N během testu paměti
- **Instalace pomocí emulátoru, neboli virtuálního PC** (VirtualBox, VMWare, Qemu) na CF kartu a následné vložení do desky ALIX.2D13

V našem případě byla zvolena instalace pomocí emulátoru **Qemu**, který je dostupný pro většinu linuxových distribucí určených pro osobní počítače jako je například Ubuntu. Zbývající dvě varianty instalace jsou popsány v referencích [27] pro PXE boot a [28] pro instalaci pomocí USB média.

Poznámka 4.1 Pro PXE boot není ze strany firmy PC Engines oficiálně poskytována technická podpora, nicméně vývojářům firmy Voyage se podařilo tuto funkci, zdá se úspěšně, implementovat.

Pro projekt HomeRouter byla vybrána edice Voyage MPD a to především z důvodu možnosti pozdější rozšiřitelnosti v komplexní síťový miniserver. Následující popis instalace Voyage Linuxu na CF kartu předpokládá osobní počítač, či notebook s OS Ubuntu, nebo Debian a čtečkou paměťových karet typu CF. Nejprve je potřeba nainstalovat samotný virtuální stroj pomocí repozitáře:

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

Výpis 4: Instalace virtuálního stroje Qemu

Nyní je potřeba stáhnout poslední dostupnou verzi LiveCD pro Voyage MPD (v našem případě je poslední dostupná verze 0.8.0) ze stránek výrobce [27] a to buď prostřednictvím webového prohlížeče, či z příkazové řádky:

```
root@ubuntu:~# wget http://www.voyage.hk/download/voyage-mpd/voyage-mpd-0.8.0.iso
```

Výpis 5: Stažení poslední verze Voyage MPD přes příkazovou řádku

Dále je potřeba zkontrolovat zda je CF karta korektně připojena a jaký je její identifikátor v OS:

```
root@ubuntu:~# fdisk -l
```

```
Disk /dev/sdb: 8019 MB, 8019099648 bytes
255 heads, 63 sectors/track, 974 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier : 0x000bfde7
```

```
Device Boot      Start         End      Blocks   Id  System
/dev/sdb1 *          1           974       7823654+  83  Linux
```

Výpis 6: Kontrola identifikátoru disku

Je zřejmé, že CF karta je v systému zařazena jako zařízení /dev/sdb. Proto již stačí jen spustit emulátor a tím započít zavádění Voyage Linuxu z LiveCD.

```
root@ubuntu:~# qemu -hda /dev/sdb -cdrom /home/ubuntu/voyage-mpd-0.8.0.iso -boot d
```

Výpis 7: Spuštění Qemu pro zavedení Voyage Linux ze staženého LiveCD



Obrázek 12: Úvodní obrazovka LiveCD Voyage Linux v emulátoru Qemu

Přepínačem *-hda* určíme CF kartu jako hlavní pevný disk virtuálního PC, přepínač *-cdrom* pak určuje cestu k iso souboru daného LiveCD. Poslední uvedený přepínač *-boot* specifikuje ze které jednotky se bude zavádět systém a jeho parametr *d* potom první dostupnou CD mechaniku. Nástroj Qemu má k dispozici mnoho dalších přepínačů upřesňujících typ hardware, který chceme emulovat, avšak to v našem případě není pro samotnou instalaci až tak podstatné a navíc by přesáhlo by to rámec této práce. Veškeré detaily jsou dostupné buď to v manuálových stránkách dané distribuce, popřípadě na oficiálních stránkách výrobce [29].

Po spuštění emulátoru Qemu se zadanými přepínači můžeme vidět úvodní obrazovku (na obrázku 12), na které zvolíme Voyage Linux. Po zavedení systému jsou výchozí přihlašovací údaje: jméno *root* a heslo *voyage*.

Po úspěšném přihlášení do systému je potřeba vytvořit distribuční složku pro instalaci a připojit image instalačního souboru do tohoto adresáře:

```
root@voyage:~# mkdir /tmp/root
root@voyage:~# mount -o loop /live/image/live/filesystem.squashfs /tmp/root
root@voyage:~# cd /tmp/root
```

Výpis 8: Instalace Voyage Linuxu na CF kartu - krok 1

Následně je nutné vytvořit přípojný bod pro instalační disk a zformátovat jej připraveným skriptem *format-cf.sh*, v našem případě je tímto diskem CF karta:

```
root@voyage:/tmp/root# mkdir /tmp/cf
root@voyage:/tmp/root# /usr/local/sbin/format-cf.sh /dev/hda
```

Výpis 9: Instalace Voyage Linuxu na CF kartu - krok 2

Ted' už je stačí jen spustit samotnou instalaci příslušným skriptem *voyage.update*:

```
root@voyage:/tmp/root# /usr/local/sbin/voyage.update
```

Výpis 10: Instalace Voyage Linuxu na CF kartu - krok 3

Při instalaci je potřeba zvolit jako distribuční adresář námi vytvořený */tmp/root* a jako přípojný bod opět předpřipravený */tmp/cf*, dále potom můžeme zvolit platformu na kterou systém instalujeme (ALIX, Soekris atd.), disk a oddíl na který chceme systém nainstalovat, tedy v našem případě */dev/sdb* a oddíl *1*, případně typ zavaděče a rychlost sériového rozhraní pro terminál. Vše je podrobně popsáno v návodu na stránkách Voyage Linux a v příloze A.1.

Po úspěšné instalaci, před samotným vložením CF karty do desky ALIX, můžeme korektní zavádění systému ověřit opět emulátorem Qemu, avšak je nutno zadat přepínač *-nographic*. Při instalaci jsme totiž zvolili platformu ALIX a Voyage Linux tedy předpokládá posílání výstupu ze zavádění systému na sériové rozhraní, resp. předpokládá připojený terminál:

```
root@ubuntu:~# qemu -hda /dev/sdb -nographic
```

Výpis 11: Ověření korektního zavádění systému

Pokud se systém korektně zavede, můžeme rovnou provést například základní konfiguraci síťových rozhraní v */etc/network/interfaces*, či nastavení samotných konfiguračních souborů Asterisku. Součástí instalace je také například editor *nano*, *openssh-server* pro okamžitý přístup po instalaci srkz protokol SSH, *openssl* pro šifrování pomocí SSL, *tcpdump* pro možnost zachytávání provozu přímo na daném rozhraní a jeho následné zobrazení například v nejrozšířenějším open-source paketovém analyzátoru Wireshark, dále *hostapd* pro vytvoření Wi-fi přístupového bodu, *ppp* a *pppoe* pro tunelování protokolu PPP přes Ethernet, skript pro nastavení NAT pomocí *iptables*, či *dnsmasq* pro rychlé nastavení DHCP/DNS serveru.

4.3 Instalace a nastavení Asterisku

Jak je typické pro operační systémy postavené na základech Debian linuxu, obvykle nejsnazší cesta k získání nového software je stažení a nainstalování z repozitáře pomocí nástroje *apt*. Instalaci asterisku tedy provedeme následovně:

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

Výpis 12: Instalace Asterisku ve Voyage Linux



Obrázek 13: Grafické rozhraní Asterisku

Tímto jsme tedy naistalovali asterisk a to aktuálně ve verzi 1.8.4.2. Po této instalaci máme v systému dostupné tyto balíky: *asterisk*, *asterisk-config*, *asterisk-core-sounds-en-gsm*, *asterisk-h323*, *asterisk-modules*. V případě distribuce Voyage ONE je mimo tyto balíky dostupný ve výchozí instalaci také balík *asterisk-gui* lehce upraven pro distribuci Voyage. V případě námi použité verze Voyage MPD lze tento balík snadno doinstalovat opět pomocí repozitáře:

```
root@voyage:~# apt-get install asterisk-gui
```

Výpis 13: Instalace grafického rozhraní pro Asterisk

4.3.1 Nastavení webového rozhraní Asterisku

Webové uživatelské rozhraní Asterisku lze vidět na obrázku 13. V případě, že se rozhodneme používat grafické rozhraní asterisku po doinstalaci přes repozitář, je potřeba provést příslušná nastavení v souboru */etc/asterisk/http.conf* kde je potřeba definovat zapnutí webového rozhraní, port na němž má služba běžet a IP adresu na níž má být toto rozhraní dostupné. Dále pak povolit statické obslužení a cestu pro statické obslužení HTTP dotazu:

```
[general]
enabled = yes
bindaddr = 127.0.0.1
bindport = 8088
enablestatic = yes
redirect = / / static /config/index.html
```

Výpis 14: Nastavení souboru *http.conf* potřebná pro správnou funkci Asterisk GUI

A taktéž v souboru */etc/asterisk/manager.conf* je potřeba dodefinovat uživatele a jejich hesla:

```
[general]
webenabled = yes

[admin]
secret = voyage
read = system,call,log,verbose,command,agent,config
write = system,call,log,verbose,command,agent,config,originate
```

Výpis 15: Nastavení v souboru *manager.conf* potřebná pro správnou funkci Asterisk GUI

Webové rozhraní asterisků může být jistě velmi vítáno nováčky, kteří nejsou alespoň základně sběhlí v konfiguraci tohoto software. Webové rozhraní s sebou totiž nese, tak jako většina webových rozhraní pro asterisk, tu nelibost, že nedokáže rozpoznat Dialplany, které již byly definovány přímou editací souboru *extensions.conf*. Tím pádem nelze přes toto webové rozhraní spravovat ani již vytvořené uživatele a další nastavení, jelikož k tomu je potřeba mít definován alespoň jeden vlastní Dialplan přes webové rozhraní. Je tedy zřejmě nepoužitelný pro pokročilejší správce Asterisků, kteří již jistě rádi používají své zaběhnuté a osvědčené Dialplany a nechtějí se zdržovat konfigurací Dialplanu klikáním přes webové rozhraní. Také z tohoto důvodu v této práci není věnováno více prostoru konfiguraci Asterisků pomocí tohoto webového rozhraní.

4.3.2 Základní konfigurace Asterisků

Nejdůležitější soubory pro základní konfiguraci Asterisků jako SIP serveru jsou *extensions.conf*, kde definujeme Dialplany, tedy pravidla jak zacházet s jednotlivými příchozími a odchozími hovory a soubor *sip.conf*, kde definujeme jednotlivé uživatelské účty. Jak bylo popsáno v kapitole Zacházení s kanály v Asterisků, Asterisk dokáže ke všem kanálům přistupovat téměř shodně a tedy pomíjí rozdíly mezi použitými technologiemi. Proto lze v *extensions.conf* definovat obecná pravidla (Dialplany) pro zacházení s kanály různých hlasových protokolů - SIP, IAX, SCCP, H.323 atd. Nicméně v tomto textu se zaměříme na konfiguraci SIP účtů a stejně tak Dialplanu pro SIP. V této hlavní části práce bude nastíněna jen základní konfigurace nutná ke spuštění SIP hovorů, tak aby bylo možno provést pozdější zátěžové testy této Asterisk PBX, popsané v kapitole 6.2. Pro editaci *sip.conf* použijeme libovolný editor dostupný ve Voyage Linux, v našem případě jsme použili editor *nano*, a upravili soubor následovně:

```

[general]
context = inner-calls
canreinvite = no

[globals]

[1000]
type=friend
context=inner-calls
secret=nejakeheslo
callerid="Vratnice" <1000>
host=dynamic
dtmfmode=rfc2833
allow=all

[1001]
type=friend
callerid="Sekretarka" <1000>
:
:

```

Výpis 16: Nastavení v souboru *sip.conf* potřebná pro základní zprovoznění SIP telefonie

Ve výpisu 16 vidíme, že v *sip.conf* jsou použity 2 sekce - *general* a *global*. V sekci *general* definujeme výchozí kontext, který bude použit pro příchozí hovory a parametr *canreinvite = no*, který udává, že veškeré RTP spojení půjdou přes Asterisk. V sekci *global* pak definujeme jednotlivé uživatelské účty kde *type=friend* říká, že následující kontext *inner-calls* definovaný v *extensions.conf* je použit pro příchozí i odchozí hovory, *secret* definuje heslo pro daného účastníka, *callerid* pak identifikaci uživatele, která se zobrazí volanému účastníkovi při příchozím hovoru. Dále *host=dynamic* znamená, že uživatel se může registrovat z libovolné IP adresy, *dtmfmode=rfc2833*, že tónová volba bude použita dle standardu rfc2833 a *allow=all* míní, že uživatel může použít jakýkoliv kodek podporován Asteriskem. Stejným způsobem nakonfigurujeme všechny ostatní účty které budeme potřebovat. Výpis konfiguračního souboru *sip.conf* je v příloze **A.2**.

Dalším, a tím nejpodstatnějším souborem pro konfiguraci je *extensions.conf*. V tomto souboru definujeme veškerá pravidla pro zacházení s hovory. Opět je zde uvedena jen základní konfigurace pro výše zmiňovaný kontext pojmenován *inner-calls* a demonstrační kontext *incoming-calls* pro ukázkou přecházení mezi kontexty:

```

[inner-calls]
;příklad 1
exten => 1000,1,Dial(SIP/1000,20,t)
exten => 1000,2,Hangup

;příklad 2
exten => _1XXX,1,Dial(SIP/${EXTEN:0},20,t)
exten => _1XXX,2,Hangup

exten => s,1,Dial(SIP/1000,20,t)
exten => s,2,Dial(SIP/1001,20,t)

```

```

exten => s,3,Hangup

[incoming-calls]
exten => s,1,Goto(inner-calls,s,1)

```

Výpis 17: Nastavení v souboru *extensions.conf* potřebná pro základní zprovoznění SIP telefonie

Příklad ve výpisu 17 ukazuje jak definovat základní pravidla pro směrování hovorů protokolu SIP. Vytvoření pravidla pro dané účastnické číslo musíme započít heslem *exten*, následuje číslo účastníka/skupiny účastníků pro kterého toto pravidlo definujeme, dále priorita zadaného pravidla a akce která se má vykonat. Z výpisu 17 je tedy v prvním řádku kontextu *inner-calls* definováno pravidlo pro účastníka *1000*, kde číslice *1* reprezentuje nejvyšší prioritu, tedy akci která se vykoná jako první pro zadaného účastníka. Heslo *Dial* s parametrem *SIP* a číslem účastníka *1000* pak vytvoří SIP spojení k účastníkovi *1000*. Další parametr *20* a *t* udávají čas v sekundách po kterém se přejde na akci s nižší prioritou. Tedy nevyzvedne-li účastník *1000* po dobu *20* sekund hovor, bude následujícím pravidlem zavěšen. Analogicky je takto možno nastavit obecné pravidlo pro všechny účty ve tvaru *1XXX* tak jak je uvedeno v příkladu 2 výpisu 17. *X* zde reprezentují číslice 0 až 9. *\$EXTEN:0* pak reprezentuje proměnnou, tedy skutečně vytočené číslo volajícím, kde nula znamená kolik pozic zleva bude z volaného čísla odebráno. Toto se hodí například pro odebrání prefixů, neboli provoleb. Například budeme-li do veřejné telefonní sítě volat přes 0, jak tomu bývá obecnou zvyklostí, musíme tuto nulu odebrat právě použitím proměnné *\$EXTEN:1*. Asterisk používá také rezervovaná jména linek (extensions names) pro speciální účely. Ve výpisu 17 je použito *s* což znamená startovací „linka“ (výchozí pravidlo). Tedy není-li volané číslo ve formátu *1XXX*, použije se toto pravidlo pro *s*. Příklad přechodu mezi jednotlivými kontexty je pak uveden v poslední sekci výpisu 17 kde je definován kontext *incoming-calls* pro příchozí hovory a v něm definované pravidlo pro přechod do kontextu *inner-calls* na řádek s pravidlem pro startovací extension *s* a prioritou *1*. Tato konfigurace je zde pouze pro ilustraci a vysvětlení jak se v *extensions.conf* s hovory zachází. Kompletní *extensions.conf* použít pro tuto práci je v příloze **A.3**.

V Asterisku je také možno tzv. monitorovat, nebo-li nahrávat hovory. Tato funkce je pro Call centra, ale i malé a střední firmy mnohdy velmi žádoucí. Proto si zde uvedeme příklad nastavení monitorování hovoru pro všechny účastníky. Toto nastavení bylo také použito pro zátěžové testy Embedded Asterisku popisovaného v této práci.

```

exten => _1XXX,1,Set(CALLFILENAME=${EXTEN:0}-${STRFTIME(${EPOCH},,%Y%m%d-%H%M%S)})
exten => _1XXX,n,Monitor(wav,/home/data/nahravky/${CALLFILENAME},mb)
exten => _1XXX,n,Dial(SIP/${EXTEN:0},25,t)
exten => _1XXX,n,Hangup

```

Výpis 18: Nastavení monitorování hovoru v souboru *extensions.conf*

V prvním řádku definujeme proměnnou *CALLFILENAME* a to tak, že její tvar bude složen posloupností: {volané číslo}-{datum ve tvaru RRRRMMDD}-{čas ve tvaru hh:mm:ss}. Ve druhém řádku definujeme samotný proces zaznamenávání hovoru a to tak,

že nejdříve definujeme formát ve kterém budeme hovor ukládat, v našem případě *wav* a cesta kam se má soubor vytvořit, pochopitelně zakončen proměnnou *CALLFILENAME* pro odlišení jednotlivých nahrávek. Výsledný soubor pak tedy bude mít například následující tvar: *1002-20120417-122549.wav*, tedy volání na číslo 1002 uskutečněno 17. dubna 2012 v čase 12:25:49.

Na konci druhého řádku výpisu 18 si můžeme všimnout paramterů *m* a *b*. Asterisk totiž přistupuje k hovoru jako ke dvěma samostatným hovorům (dvěma různým RTP tokům), tedy směr volající-volaný a volaný-volající. Asterisk tedy potom uloží do souboru ve tvaru proměnné *CALLFILENAME* s dovětkem *-out* hlas volajícího (odchozí RTP stream) a naopak do souboru ve tvaru proměnné *CALLFILENAME* s dovětkem *-in* uloží hlas volného (příchozí RTP stream). K tomu abychom mohli tyto dva soubory po ukončení hovoru sloučit v jeden nám slouží nástroj *sox* (v některých distribucích nazýván také *soxmix*). A právě parametr *m* zadán na konci druhého řádku výpisu 18, zajistí provedení této konverze v jeden soubor po ukončení hovoru. Zbývající parametr *b* definuje započtení nahrávání až v momentě sestavení obousměrného RTP toku, tedy až po vyzvednutí hovoru volaným.

Další možností je tzv. podmíněné nahrávání. Uživatel si sám může zvolit kdy chce mít hovory monitorovány a kdy ne. V příloze **A.5** je ukázka podmíněného nahrávání. Účastík si může aktivovat funkci monitorování hovoru vytočením **7** a naopak deaktivovat pomocí ***7*. Není-li si účastník jistý nastavením své volby, může si jí snadno ověřit pomocí volby **77* kdy mu Asterisk přehraje informační zprávu o stavu monitorování jeho hovorů. Obdobně můžeme nastavit podmíněné přesměrování hovorů atd. Tato nastavení už by ale přesáhla rámeček této práce. Pro podrobnější informace o nejrozličnějších nastaveních Asterisku jsou vhodné webové stránky www.voip-info.org [31].

5 Posouzení energetické náročnosti a porovnání se současnými systémy

Jak bylo uvedeno v podkapitole 3.2 množství energie spotřebované na samotný provoz serverů a datových center v rámci celého internetu tvoří jedno procento globální spotřeby elektrické energie. Proto je velmi důležité zaměřit se na samotnou spotřebu síťových prvků a systémů provozovaných například v rámci určité firmy. Spotřeba běžně používaných PC a serverů je velmi závislá na přesné konfiguraci daného PC/serveru. Nicméně ve velmi obecné rovině lze předpokládat průměrnou spotřebu stolního PC bez připojených periférií okolo 100 W. Jako příklad může posloužit konfigurace v tabulce 10, počítáno s maximální možnou spotřebou všech komponent [32]:

Komponenta	Spotřeba [W]
základní deska	23
CPU (Athlon A64 3000+)	64
grafická karta (GeForce FX-5200)	25
paměť (512 MB DDR 400)	10
zvuková karta (SB Live! 5.1)	8
disk (SEAGATE Barracuda 7200.7 80GB)	13
DVD-RW	25
Celková spotřeba	168

Tabulka 10: Spotřeba průměrné PC sestavy při plném zatížení

Tento součet příkonů je pochopitelně velmi orientační a těžko předpokládat, že naše PC/serverová sestava poběží 24 hodin denně v plném zatížení. Proto byla pro tuto diplomovou práci změřena výkonostně průměrná sestava popsána v následující podkapitole.

5.1 Porovnání spotřeby vzorové PC sestavy a desky ALIX.2D13

V této práci byla jako demonstrativní sestava pro porovnání energetické náročnosti použita sestava z tabulky 11.

Komponenta	Spotřeba [W]
základní deska Gigabyte GA-P41T-D3P	
CPU (Intel Celeron E3300)	
grafická karta (ATI RAdeon 9600)	
paměť (2GB DDR 400)	
zvuková karta (integr. Realtek HD audio)	
disk 1 (SEAGATE Barracuda 7200 ot. 500GB)	
disk 2 (SEAGATE Barracuda 7200 ot. 80GB)	
DVD-RW Sony	
Celková naměřená spotřeba	112

Tabulka 11: Spotřeba měřené PC sestavy při cca. 20% zatížení CPU

Měření bylo provedeno pomocí běžně dostupného komerčního měřiče spotřeby, přičemž na sestavě běžel operační systém Ubuntu 11.04 s Asteriskem ve verzi 1.8.4.2. Vytížení procesoru na 20% bylo dosaženo pomocí několika současných hovorů generovaných nástrojem SIPP a současně zaznamenávaných na disk. Použitý měřič spotřeby elektrické energie má pochopitelně značnou odchylku, jelikož jak se zdá předpokládá čistě odporovou zátěž, tedy účinník $\cos \phi = 1$.

Nicméně, změřili-li jsme totožným měřičem spotřeby Embedded Asterisk PBX s dvěma připojenými USB disky o velikostech 500 a 1000 GB a současně zaznamenávali hovory na větší z těchto dvou disků. Naměřené hodnoty jsou v tabulce 12.

Komponenta	Spotřeba [W]
ALIX.2D13	
Celková naměřená spotřeba	15

Tabulka 12: Spotřeba Embedded Asterisk PBX při cca. 20% zatížení CPU

Porovnáme-li tedy výše naměřené hodnoty, je zřejmé, že Asterisk běžící na ALIX.2D13 dosahuje bezmála **desetinové** hodnoty spotřeby elektrické energie oproti Asterisku běžícímu na výše uvedené sestavě v tabulce 10. Navíc je potřeba počítat s určitou účinností zdroje jež ALIX.2D13 napájí, jelikož v praxi může být deska ALIX napájena přes pasivní PoE, či ze záložního zdroje UPS, popřípadě i z baterie. V našem případě byl použit stabilizovaný zdroj výrobce Sunny s výstupním napětím 18 V (SS) a maximálním možným dodávaným proudem 1,33 A. Rodina těchto levných minimalizovaných stabilizovaných zdrojů má obecně účinnost okolo 80%. Tedy spotřeba samotné desky ALIX.2D13 je pak namísto 15 W přibližně **12 W**. Bez připojených USB zařízení je pak tato hodnota cca. **8,8 W**. Výrobce PC Engines udává spotřebu **3-4 W** jen při běhu OS a cca. **6 W** při spuštěných ostatních procesech bez připojené miniPCI karty a USB zařízení.

5.2 Kalkulace návratnosti investice do Embedded Asterisk PBX

Zohledníme-li tedy výše naměřené hodnoty můžeme si snadno spočítat návratnost investice do Embedded Asterisk PBX. Zohledníme-li navíc, že se nemusí jednat pouze provozování PBX na zakoupeném hardware, ale také provozování ostatních síťových služeb, pak je tato myšlenka kalkulace ještě zajímavější. Nejdříve si tedy zrekapitulujme ceny jednotlivých komponent potřebné pro vytvoření komplexního miniserveru Home-Router provozujícího Asterisk PBX, který je zobrazen na obrázku 14. Ceny jednotlivých komponent v tabulce 13 jsou kalkulovány k 17.4.2012 z ceníků [33], [34].

Komponenta	Cena (vč. DPH)
ALIX.2D13	2 685,- Kč
Kryt desky	265,- Kč
Compact Flash 4GB	209,- Kč
Napájecí zdroj 12V/1,5A	156,- Kč
Celková cena	3 315,-Kč

Tabulka 13: Ceny komponent potřebných ke kompletu Embedded Asterisk PBX



Obrázek 14: Výsledek práce - Embedded Asterisk PBX

Pokud tedy vezmeme v úvahu variantu, že ve firmě již nějaký, například poštovní server provozujeme, popřípadě máme v plánu Asterisk zprovoznit na starším „vyřazeném“ PC, jinými slovy nebudeme počítat s pořízením nového hardware jako alternativy k ALIX.2D13, nýbrž zohledníme pouze spotřebu takového serveru, pak snadno spočítáme návratnost investice do pořízení miniserveru HomeRouter. Budeme-li předpokládat jednotarifní odeběr energie od našeho distributora, tak jak se ve většině domácností a malých firem děje, dostaneme cenu okolo **5 Kč** za kWh jako cenu koncovou, vč. distribučních poplatků. Z naměřených hodnot v tabulce 11 je zřejmá spotřeba **112 W** pro testovanou PC sestavu a z tabulky 12 potom **15 W** pro desku ALIX.2D13. Pak tedy v tabulce 14 dostáváme porovnání cen za spotřebovanou energii kalkulovanou pro provoz 365 dní v roce:

Sestava	Spotřeba za rok [kWh]	Cena (vč. DPH)
ALIX.2D13	131,4	657,- Kč
Testovaná sestava	981,1	4 905,- Kč

Tabulka 14: Porovnání finanční náročnosti na spotřebovanou el. energii

Z výše uvedené tabulky 14 je zřejmé, že návratnost investice do Embedded Asterisk PBX je **méně než jeden rok**, což v myšlence Green Computing hraje velmi významnou roli.

6 Zhodnocení dosažených výsledků

Z výše popsaných vlastností a funkcí této Embedded Asterisk PBX tedy vyplývá, že jsme dostali komplexní miniserver s operačním systémem Voyage Linux, na kterém lze provozovat libovolnou síťovou službu a nikoli pouze samotný Asterisk PBX. Nicméně zaměření na VoIP telefonii a tedy samotný Asterisk PBX, bylo majoritním úkolem této diplomové práce. K tomu abychom mohli odpovědně vyhodnotit užitečnost takovéto PBX a tedy i její samotnou použitelnost v SOHO segmentu trhu, bylo potřeba provést několik testů zajišťujících objektivní hodnocení tohoto produktu. Všechny tyto testy jsou popsány v následujících podkapitolách.

6.1 Měření datové propustnosti Embedded Asterisk PBX

Prvním testem a současně základním parametrem pro samotný chod Asterisk PBX, je jistě datová propustnost, kterou je schopna deska ALIX.2D13 s OS Voyage Linux zvládnout. Výrobce udává propustnost všech tří síťových rozhraní až 100 Mbps. Samozřejmě tyto hodnoty jsou pouze teoretické a v oblasti embedded systémů často zavádějí. Proto byl proveden test měření datové propustnosti Embedded Asterisk PBX nástrojem IPERF. Nástroj IPERF je jedním z mála spolehlivých a objektivních nástrojů pro měření TCP a UDP datové propustnosti a je také např. jediným open-source nástrojem který je uznáván ve společnosti VerizonBusiness, která je majoritním poskytovatelem MPLS VPN v celém světě. Samotné měření bylo provedeno připojením dvou PC v níže popsaných konfiguracích pomocí ethernetových kabelů délky cca. 1,5 m. Vzhledem k tomu, že TCP protokol je spojově orientován, délka ethernetového kabelu, či samotná síťová infrastruktura přes kterou datovou propustnost měříme a tedy i RTD jsou hlavní faktory ovlivňující výslednou TCP propustnost. Dále je propustnost závislá také například na HW konfiguraci koncových zařízení, v našem případě PC, a aktuální vytiženosti jejich síťových rozhraní. Proto byly testy prováděny v klidových hodnotách využití CPU a síťových karet obou koncových stanic. Hardwarové konfigurace koncových stanic jsou popsány v tabulce 15.

CPU	Počet jader CPU	RAM	Rychlost síř.karty
Intel Atom 1,6 Ghz	jedno	1 GB	1000/100 Mbps
Intel Centrino 1,6 Ghz	dvě	1 GB	1000/100 Mbps

Tabulka 15: Hardwarová konfigurace koncových stanic pro měření datové propustnosti Embedded Asterisk PBX

Vzhledem k tomu, že byla uvažována možnost budoucí rozšiřitelnosti Embedded Asterisk PBX o další služby a tedy i funkci směrovače, byly testy prováděny jak v rámci jedné podsítě, tedy přepínání paketů na L2 mezi dvěma rozhraními Embedded Asterisk PBX a také se současným směrováním mezi dvěma podsítěmi, tedy směrování na L3 RM OSI modelu. Výsledky byly velmi překvapující a od teoretických hodnot uváděných výrobcem se příliš nevzdalovaly - viz výpisy 19 a 20.

```

root@ubuntu:~# iperf -c 192.168.3.6 -m
-----
Client connecting to 192.168.3.6, TCP port 5001
TCP window size: 16.0 KByte (default)
-----
[ 3] local 192.168.3.8 port 54887 connected with 192.168.3.6 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.0 sec 112 MBytes 94.5 Mbits/sec
[ 3] MSS size 1448 bytes (MTU 1500 bytes, ethernet)

```

Výpis 19: Měření TCP propustnosti v rámci jedné podsítě

```

root@ubuntu:~# iperf -c 192.168.4.6 -m
-----
Client connecting to 192.168.4.6, TCP port 5001
TCP window size: 16.0 KByte (default)
-----
[ 3] local 192.168.3.8 port 54887 connected with 192.168.4.6 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.0 sec 112 MBytes 94.2 Mbits/sec
[ 3] MSS size 1448 bytes (MTU 1500 bytes, ethernet)

```

Výpis 20: Měření TCP propustnosti mezi dvěma podsítěmi

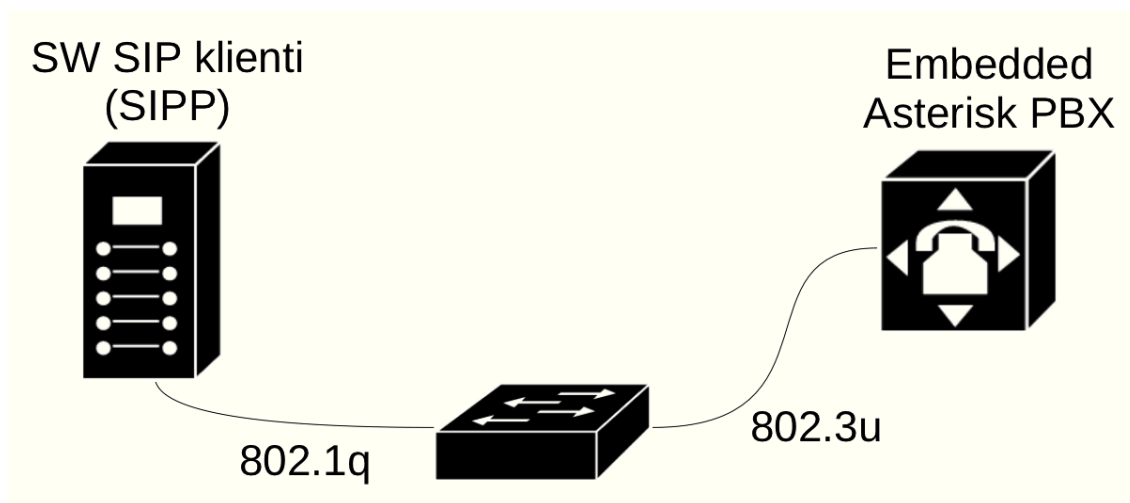
Obdobně bylo provedeno měření datové propustnosti při použití nespojově orientovaného transportního protokolu UDP, přičemž rozdíly ve výsledcích byly minimální - **95,5 Mbps** v rámci jedné podsítě a **95,1 Mbps** mezi dvěma podsítěmi.

6.2 Měření počtu současně obslužených SIP hovorů

Tím hlavním výkonnostním parametrem pro samotnou Embedded Asterisk PBX a VoIP telefonii obecně, je jistě počet hovorů terý takováto PBX zvládne obsloužit současně. Jak ze znalosti protokolů SIP a RTP plyne, je jistě velkým rozdílem zda Embedded Asterisk obsluhuje pouze SIP relace, tedy zprávy pro vytvoření a ukončení hovoru, nebo jsou přes něj směrovány také samotné hovory, tedy režim kdy RTP pakety procházejí skrz Asterisk. Toto je výhodné pro mnohé doplňkové služby jako je například monitorování, či nahrávání hovorů které jsou zákazníky čím dál častěji vyžadovány. Proto v námi prováděných testech byly testovány následující situace:

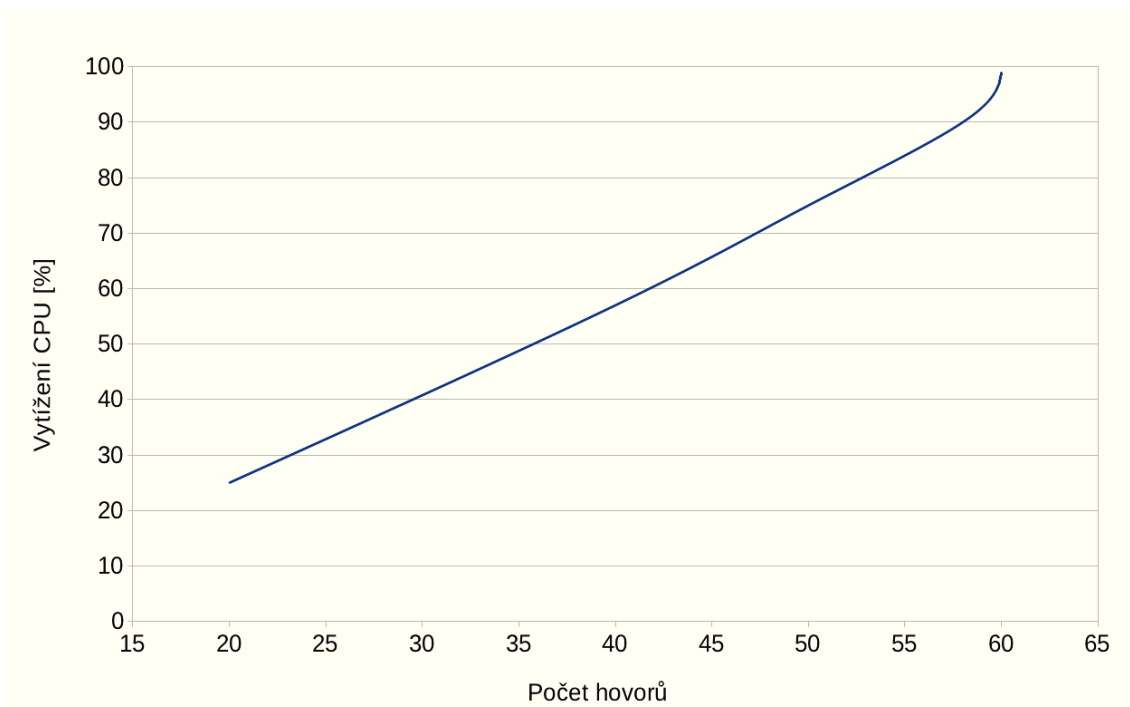
- **Počet obslužených SIP spojení s procházejícími RTP pakety skrz Asterisk**
- **Počet obslužených SIP spojení se současným záznamem uskutečněných hovorů**

Samotné testování bylo prováděno nástrojem **SIPP**, který je ostatně pro testování Asterisku doporučován i jeho autorem firmou Digium. Nástroj SIPP dokáže simulovat tisíce současných hovorů a to jak samotné SIP relace, tak se současnými RTP paktety kdy simuluje reálné hovory. Pro testování bylo využito konfiguračních souborů, které jsou uvedeny v přílohách **A.2**, **A.3** a **A.4**.



Obrázek 15: Testování Asterisk PBX nástrojem SIPP

V prvním ze zmíněných testů byl asterisk podroben postupné zátěži pomocí paralelních hovorů generovaných nástrojem SIPP (s použitím speciálního skriptu psaného v programovacím jazyku *Bash*). Při tomto testu bylo sledováno vytížení procesoru pomocí nástroje *top* a současně byly sbírány statistiky o SIP zprávách. Pomocí zmíněného skriptu byly každou vteřinu odeslány dvě žádosti o spojení, tedy dvě zprávy INVITE. Test byl prováděn s použitím čtyř virtuálních rozhraní konfigurovaných na PC na němž byl nástroj SIPP instalován a tedy prováděn test. Současně bylo využito síťové infrastruktury se switchem výrobce Cisco podporujícím enkapsulaci 802.1q. Názorné schéma zapojení je patrné z obrázku 15. Zmiňovaný skript tedy na dvou virtuálních rozhranních inicioval hovory zprávou INVITE a na zbývajících dvou virtuálních rozhranních na tyto zprávy odpovídal, tedy sestavoval reálné SIP hovory. Jako náplň hovoru (RTP streamu) byl použit předem namluvený soubor ve formátu *wav*, na jehož základě poté byla vyhodnocována kvalita současných hovorů při testu s aktivním nahráváním hovorů. Současně byl monitorován procesor pomocí nástroje *top* přes terminálové připojení k testované ústředně. Výpis nástroje *top* je zobrazen ve výpisu 21. Výše popsáním způsobem byly empiricky přidávány hovory až do nalezení hraniční hodnoty. Jako hraniční se ukázalo být **50 současných hovorů** kdy byl ještě Asterisk schopen bez problémů obsloužit všechny hovory, tedy tak aniž by docházelo k prodlevám a následné expiraci doby pro očekávanou odpověď (time-out) ze strany serveru. Vytížení CPU při tomto počtu hovorů se pohybovalo okolo **75%** a zkonsumovaná operační paměť byla **13,1%**, tedy **33,5 MB**. Jako maximální únosné zatížení bylo stanoveno 58 současných hovorů, kdy ještě došlo k úspěšnému ukončení všech současných spojení, avšak se značným zpožděním. Při tomto počtu hovorů již totiž docházelo k retransmisím zprávy BYE, tedy klient zaslal zprávu BYE, ale nepřišla mu ve stanoveném intervalu odpověď 200 OK a klient tedy zaslal zprávu BYE znovu. Všechny uskutečněné hovory nakonec sice byly ukončeny úspěšně, avšak časová prodleva již byla značná. Při 60 současných hovorech už potom proces



Obrázek 16: Graf vytížení CPU v závislosti na počtu současných hovorů

asterisk využil veškerý dostupný výpočetní výkon, CPU využíval na 99,9%, a následně došlo ke krachu celého systému a jeho samovolnému restartu pomocí funkce Watch-Dog, kterou Voyage Linux disponuje - takovéto hraniční vytížení je patrné právě z výpisu 21. Graf vytížení CPU v závislosti na počtu současných hovorů pak můžete vidět na obrázku 16. Z grafu je tedy patrné, že křivka vytížení CPU je téměř lineární, a je zde tedy lineární závislost mezi počtem hovorů a procentuálním vytížením CPU. Pro tento první test bez nahrávání hovorů byl použit konfigurační soubor *extensions.conf* uveden v příloze A.3.

```

root@voyage:~# top -b | grep 3205
  PID USER   PR NI  VIRT RES  SHR S  %CPU %MEM  TIME+  COMMAND
 3205 asterisk -11  0 77276 32m 10m  S   95.8  13.1   6:35.54 asterisk
 3205 asterisk -11  0 77276 32m 10m  S   93.8  13.1   6:38.38 asterisk
 3205 asterisk -11  0 77276 32m 10m  S   95.5  13.1   6:42.18 asterisk
 3205 asterisk -11  0 77276 32m 10m  S   99.9  13.1   6:45.98 asterisk

```

Výpis 21: Měření CPU nástrojem *top*

Nástroj *top*, který je součástí téměř každého linuxového jádra, nám poskytuje informace o aktuálně běžících procesech. Je ovšem třeba vzít v potaz, že samotný proces *top* vytěžuje do určité míry procesorovou jednotku, když se snaží získat informace o jejím vytížení (v našem případě byl interval vyčítání statistik nastaven na 3 vteřiny). Je tedy zřejmé, že procesor byl zcela zahlcen v momentě, kdy jej *asterisk* vytěžoval na 99,9%,

přičemž minimálně proces *top* a velmi pravděpodobně i ostatní procesy v systému, si žádaly v té době obsluhu od procesorové jednotky. Jednotlivé sloupce ve výpisu 21, významné pro naše testování, mají následující význam. PID udává identifikační číslo procesu, druhý sloupec *USER* vypisuje identitu uživatele pod níž daný proces běží, třetí sloupec *PR* vypisuje aktuální výši priority daného procesu, sloupec *NI* udává výši priority zadanou příkazem *nice* (záporná čísla udávají vyšší prioritu, než kladná), dále potom sloupec *%CPU* udává procentuální vytiženost procesoru příslušným procesem a sloupec *%MEM* zase procentuální vytiženost operační paměti tímto procesem.

Jak bylo zmíněno u popisu sloupce *NI*, prioritu procesu je možno v linuxu měnit pomocí příkazu *nice*, resp. *renice*. Příkazem *nice* však lze měnit prioritu pouze u spouštěného procesu, nikoli tedy u procesu, který již běží a byl například spuštěn při startu systému, tak jak je tomu i v případě procesu *asterisk*. K tomu nám tedy slouží příkaz *renice*, jehož syntaxe je uvedena ve výpisu 22. Jak již bylo zmíněno, záporná čísla udávají vyšší prioritu, než kladná a proto je v tomto příkladu zvýšena priorita procesu *asterisk* na hodnotu 18 z 20-ti možných.

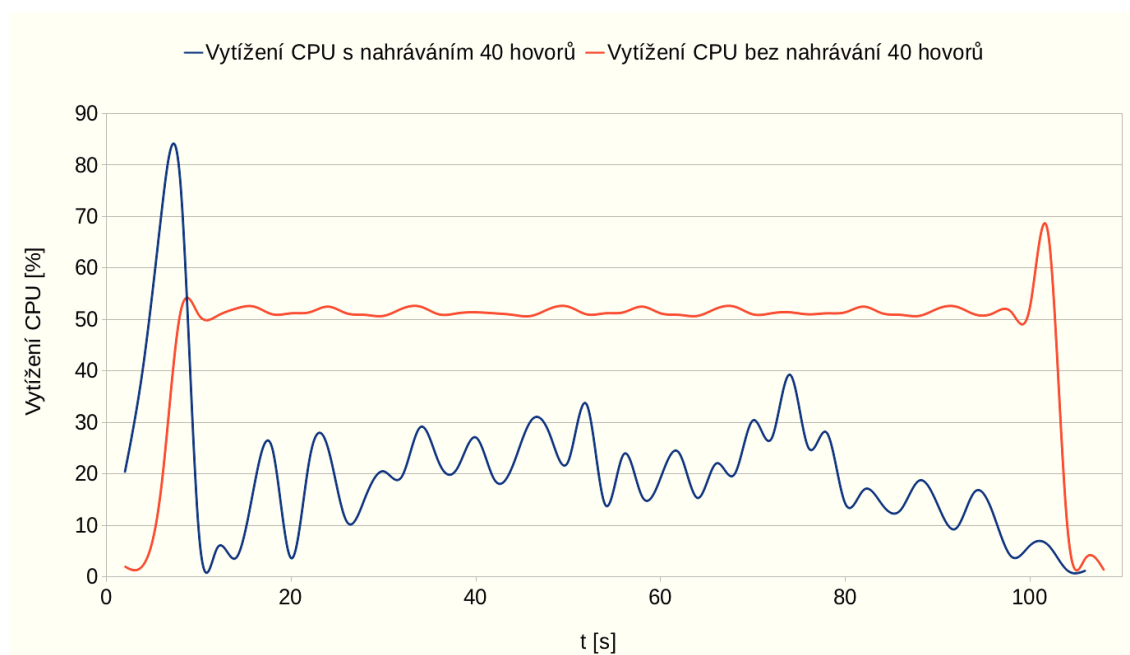
```
root@voyage:~# renice -18 3205
```

```
3205: old priority 0, new priority -18
```

Výpis 22: Zvýšení priority procesu *asterisk*

Běžný uživatel smí pomocí příkazu *nice*, resp. *renice* měnit prioritu pouze procesů jejichž je vlastníkem a navíc ji smí pouze snižovat. Pak tedy takovéto snížení priority procesu má význam pouze v případě, že aktuální uživatel má spuštěno více procesů pod svým uživatelským jménem. Uživatel *root* pak pochopitelně může měnit prioritu všech běžících procesů nezávisle na vlastníkově procesu a to jak směrem dolů, tak i nahoru. Při námi aplikovaných testech se ukázal být vliv zvýšení priority běžícího procesu zcela zanedbatelný co do nárůstu počtu současně obslužených hovorů a to především proto, že pouze proces *asterisk* má již od svého spuštění při zavádění systému nastavenou výchozí hodnotu *-11* (patrné z výpisu 21) a tedy značně vyšší než ostatní spuštěné procesy. Změna priority procesu se pak tedy uplatní pouze v případě, že Embedded Asterisk PBX provozujeme například právě jako komplexní minimalistický systém HomeRouter, jež obsluhuje více síťových aplikací současně [35].

Druhý ze zmíněných testů, tedy test kdy Asterisk současně zaznamenává hovory na připojený USB disk, byl prováděn zcela stejným postupem, jen s tím rozdílem, že byl pozměněn konfigurační soubor *extensions.conf* - viz příloha A.4. Bylo tedy aktivováno nahrávání všech současných hovorů na stejný disk. V tomto případě je zcela zřejmé, že počet současných hovorů je limitován nejen samotným vytižením CPU jednotky procesem *asterisk*, ale především režijními náklady spotřebovanými na obsluhu zápisu na paměťové médium (počet obsluh přerušení) a tedy i rychlostí samotných sběrnic a také připojeného USB média. Při prováděných měřeních CPU nástrojem *top* bylo zjištěno, že například při 40-ti současných hovorech, které ještě byly úspěšně obslouženy bez retransmisí zpráv *BYE*, bylo vytižení CPU procesem *asterisk* velmi kolísavé - viz obrázek 17. Toto bylo zřejmě způsobeno právě obsluhou přerušení od USB řadiče, které zkonzu-



Obrázek 17: Vytížení CPU procesem *asterisk* při 40-ti současných hovorech bez a s nahráváním hovorů

movalo určitý výkon CPU na samotný zápis. Také ze zaznamenávaných souborů, pak bylo zřejmé přerušování daného RTP toku, které dosahovalo zkrácení nahraného souboru například až o dvacet vteřin. Pro lepší vyhodnocení byla použita nahrávka v niž počítáno od 1 do 1000, přičemž výše zmíněným skriptem přednastavená délka hovoru byla 100 sekund, tedy 1:40. Některé výsledné nahrávky pak byly zkráceny například až na 1:20, tedy téměř pětina zaznamenávaného hovoru byla ztracena.

Jako reálná hranice maximalního počtu hovorů, které lze současně zaznamenávat na připojený USB disk bylo stanoveno **20 současných hovorů** s aktivním monitorováním. Při tomto počtu hovorů ještě docházelo ke korektnímu záznamu hovorů a zaznamenané hovory obsahovaly souvislé počítání obsažené ve výše zmíněné nahrávce, bez přerušení.

6.3 Možnosti rozšíření Embedded Asterisk PBX na miniserver HomeRouter

Výsledkem této diplomové práce je minimalizovaná pobočková ústředna Asterisk pracující na OS Voyage Linux. Tím, že Voyage Linux je derivátem Debian Linuxu (jak bylo popsáno v kapitole 4.1) a je na něj tedy možné instalovat libovolné *dpkg* balíčky, možnosti jeho rozšiřitelnosti jsou v podstatě neomezené, resp. omezené jen na množinu těchto *dpkg* balíčků vyvíjených pro systém Debian. Navíc jasným záměrem této práce bylo navrhnout a zkompletovat zařízení určené do SOHO segmentu trhu, tedy tam kde je preferována spíše jednotnost, než rozdělenost. Tedy pro koncového zákazníka bude jistě větším lákadlem pořídit si zařízení na němž bude moci provozovat všechny své síťové služby, než několik různých zařízení pro každou službu odděleně. Jak bylo uve-

deno v popisu desky ALIX.2D13 (kapitola 4), tato deska disponuje slotem miniPCI pro možnost připojení Wi-fi karty, kterou lze následně použít jak v módu přístupového bodu (AP), tak jako klienta. Tímto tedy můžeme nahradit další síťový prvek, ačkoli z hlediska síťového designu pro kampusy, či pobočky (tzv. Branch Office) to není zcela ideální, jelikož přístupová vrstva (Access Layer) není oddělena od vrstvy distribuční (Distribution Layer). Toto ovšem z pohledu designu síťové infrastruktury, v rozměru SOHO segmentu, nehraje až tak významnou roli, jelikož předpokládáme obecně systém i síť užívanou spíše jednotkami, maximálně desítkami uživatelů. Všechny níže zmíněné rozšíření lze snadno doinstalovat z repozitáře pomocí již dříve zmiňovaného příkazu: `apt-get install {název_balíku}`.

6.3.1 Rozšíření o funkci přístupového bodu - AP

K provozování obecného linuxového systému jako AP je v praxi nejčastěji používaným nástrojem, napříč všemi linuxovými distribucemi, balík *hostapd*. Tento balík je obsažen v distribuci Voyage Linux již po jeho instalaci a není tedy potřeba jej doinstalovat z repozitáře. Konfigurace balíku *hostapd* je velmi závislá na typu použité Wi-fi karty, resp. čipsetu, který tato karta používá. Proto je potřeba nejprve pečlivě vybrat typ podporovaného čipsetu a následně vhodného výrobce. Pro námi demonstrováný případ byla použita karta označovaná jako CM9 pracující v pásmech definovaných standardy 802.11a/b/g, která využívá již dlouhodobě ověřeného čipu Atheros AR5213A. V případě kompletovaného miniserveru HomeRouter je žádoucí funkce přístupového bodu v pásmech 802.11g, směrovače i prepínače současně. Po vložení zmíněné wi-fi karty do miniPCI slotu desky ALIX.2D13 je tato karta obvykle detekována jako rozhraní *wlan0*. Toto lze snadno ověřit pomocí příkazu *iwconfig* - viz výpis 23.

```
root@voyage:~# iwconfig wlan0
wlan0 IEEE 802.11abg Mode:Master Frequency:2.437 GHz Tx-Power=27 dBm
      Retry long limit :7   RTS thr:off   Fragment thr:off
      Power Management:on
```

Výpis 23: Bezdrátové rozhraní *wlan0* po zapojení karty CM9

V příkladové konfiguraci je vytvořen most (tzv. Bridge mode) mezi dvěma z ethernetovými rozhraními *eth1* a *eth2* a rozhraním bezdrátové karty wi-fi *wlan0*. K docílení této konfigurace je potřeba modifikovat soubory */etc/network/interfaces* pro nastavení bezdrátového rozhraní a */etc/hostapd/hostapd.wlan0.conf* pro nastavení rozhraní *wlan0* jako AP a jeho názvu sítě (SSID), operačního módu wi-fi karty - 802.11a/b/g, pracovního kanálu a typu použitého šifrování. Po nastavení zmiňovaných souborů dle přílohy A.6 dostaneme přístupový bod s názvem: *HomeRouter* a přístupovým heslem: *voyage-linux* podporující šifrování WPA-TKIP a pracující na kanálu 6 standardu 802.11g.

6.3.2 Rozšíření o funkci DHCP serveru

Protokol DHCP je protokolem pro automatickou konfiguraci IP v klientských stanicích, či obecných zařízeních podporující IP a protokol DHCP. Tento protokol je v posledních

letech hojně používaným a to především v LAN sítích. Pro realizaci služby DHCP na HomeRouter je použit balík dostupný již po instalaci samotného OS Voyage Linux a to balík s názvem *dnsmasq*. Samotná konfigurace se pak provádí v souboru */etc/dnsmasq.conf*. Jeho příkladová konfigurace je v příloze A.7. Tato konfigurace je opět pouze základní, jelikož nastavení balíku *dnsmasq* skýtá mnoho možností vč. podpory mnoha DHCP-OPTION, domény atd. Pro detailní nastavení doporučuji důkladně pročíst výchozí konfigurační soubor dostupný v systému po instalaci.

6.3.3 Rozšíření o funkci FTP serveru

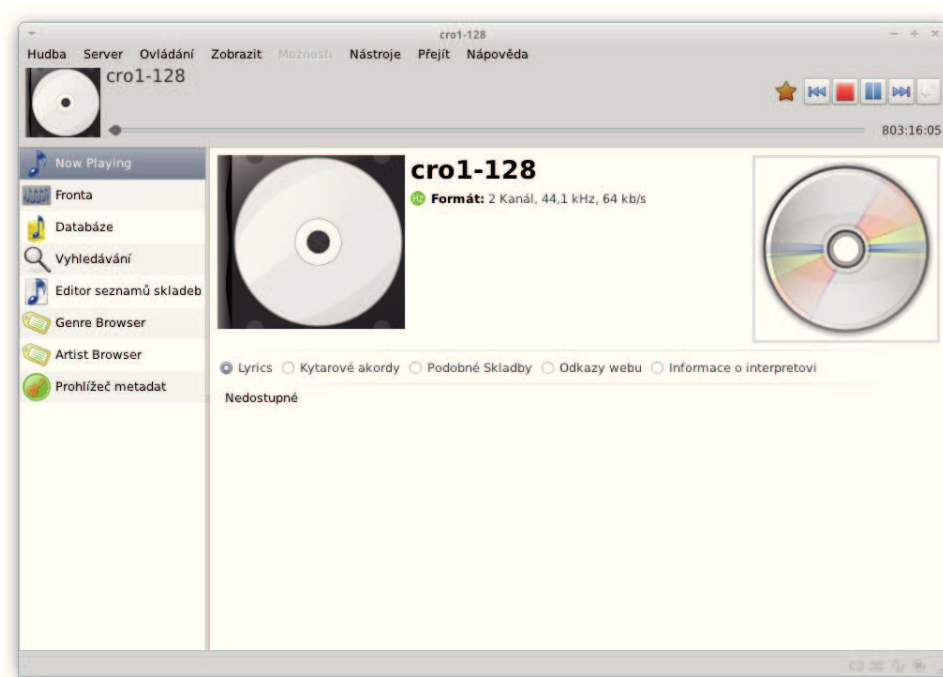
Pro funkci miniserveru HomeRouter jako serveru FTP byl vybrán balík *vsftpd*. Jak samotná zkratka vsftpd (Very Secure File Transfer Protocol Daemon) napovídá jedná se o FTP server, který klade důraz na bezpečnost. Tím je míněno, že například nepovolí (ve výchozí konfiguraci) připojení jiného, než v systému existujícího uživatele. Konfigurace balíku je realizována přes soubor */etc/vsftpd.conf* a jeho příkladná konfigurace je opět v příloze - A.8. Dle zmiňované konfigurace v příloze je povoleno anonymní přihlášení po němž je uživateli zobrazen obsah složky */sro/ftp/*. Toto umístění lze změnit přidáním parametru *local_root* do konfiguračního souboru */etc/vsftpd.conf*, a to např. takto *local_root = /home/anonym*. Tímto jsme změnil výchozí složku pro anonymního uživatele a měli bychom jí také nastavit patřičná práva pro čtení a zápis, které chceme anonymním uživatelům povolit. Uživatelé, kteří existují v systému jsou vždy po přihlášení směřováni do své domovské složky umístěné v adresáři */home/jméno_uživatele*.

6.3.4 Rozšíření o funkci HTTP serveru

Pro funkci HTTP serveru byl vybrán odlehčený balík *lighttpd*. Jedná se o velmi rychlý HTTP server s minimálními nároky na operační paměť. V případě miniserveru HomeRouter konzumuje pouze 0.6% z celkové operační paměti. Jeho nastavení se provádí opět pomocí konfiguračního souboru umístěného v */etc/lighttpd/lighttpd.conf*, přičemž příklad nastavení tohoto souboru je uveden v příloze A.9. Tento balík podporuje také MySQL databázové dotazy. Domovský adresář pro umístění webových stránek lze zvolit parametrem *server.document-root = "/var/www"*, kde */var/www* je cesta k HTML/MySQL souborům.

6.3.5 Rozšíření o funkci SMB serveru

Protokol SMB (Server Message Block), také znám jako CIFS (Common Internet File System) je dnes stále nejpoužívanějším protokolem pro lokální sdílení souborů a tiskáren v počítačových sítích typu LAN a to zejména díky jeho implementaci v OS Microsoft Windows. Svobodnou implementaci protokolu SMB přináší projekt SAMBA a jeho stejnojmenný balík dostupný opět z repozitáře. Nainstalováním zmíněného balíku *samba* získáme SMB server s nemalými možnostmi konfigurace. Opět je uvedena pouze vzorová konfigurace v příloze A.10.



Obrázek 18: GMPC - Klient pro ovládání MPD serveru

6.3.6 Rozšíření o funkci UPnP/DLNA serveru

Protokol UPnP, dnes zašticován aliancí DLNA, zažívá v posledních letech období rozkvětu. Jedná se o protokol pro sdílení médií, avšak s tím rozdílem oproti protoklu SMB, že uživatel nemá přímý přístup k úložišti dat a je tedy vhodný ke sdílení i ve veřejné síti Internet. Vše je realizováno přes protokol HTTP a server tedy klientovi pouze vystaví HTTP link pomocí něhož si uživatel může přehrát zmiňovaný multimediální soubor tedy audio, video, či foto. Dnes již i cenově průměrné TV disponující Ethernetovým portem obvykle tento protokol podporují a je tedy možno prostřednictvím LAN přehrávat multimédia uložená na miniserveru HomeRouter skrz protokol UPnP v domácí, či firemní TV. Navíc umožňuje protokol UPnP také vzdálené ovládání zařízení přes lokální síť, tedy je možno například ovládat Vaši TV, disponující podporou UPnP, pomocí mobilního telefonu např. s OS Android. Pro projekt HomeRouter byl vybrán balík *mediatomb* a to především díky širokým možnostem konfigurace. UPnP je sice zašticován aliancí DLNA, avšak i přestože se výrobci k dodržování tohoto standardu hlásí, často jej nedodržují a je tedy potřeba přidat, či odebrat jisté položky z HTTP záhlaví pro vystavování HTTP odkazů klientům, tak aby byli schopni multimédia úspěšně přehrát, resp. zobrazit. K těmto změnám slouží právě konfigurační soubor balíku *mediatomb* umístěn v `/etc/mediatomb/config.xml`. Příklad konfiguračního souboru, upraveného tak, aby *mediatomb* úspěšně komunikoval mimo OS MS Windows a Ubuntu/Debian také s TV výrobce Samsung, je

uveden v příloze A.11. Management sdílených souborů dle zmíněného konfiguračního souboru je pak možno provádět přes webové rozhraní na adrese: <http://192.168.3.1:49152>.

6.3.7 Rozšíření o funkci MPD serveru

Zkratka MPD znamená Music Player Daemon. Stejnojmenný balík umožňuje realizovat síťový přehrávač hudby přímo na miniserveru HomeRouter a to díky připojené USB zvukové kartě. V projektu HomeRouter byla použita externí USB zvuková karta AXAGO ADA-X5 využívající ovladače Alsá. Vzhledem k tomu, že se jedná o zvukovou kartu disponující až 7.1 kanálovým poslechem, její nastavení je možno provést pomocí příkazu *alsamixer*, kde můžeme definovat nastavení úrovní hlasitostí pro jednotlivé vstupy a výstupy. Samotný MPD pak konfigurujeme pomocí souboru */etc/mpd.conf*. Jeho příklad je uveden v příloze A.12. V tomto souboru je nutno definovat především parametry *music_directory* pro výchozí knihovnu, *bind_to_address* a *port* pro specifikaci IP adresy a portu na kterém bude MPD server naslouchat. Dále potom nutno specifikovat parametrem *audio_output*, kterou zvukovou kartu v systému má MPD používat. Pro management knihovny a samotné přehrávání zvukových souborů je pak potřeba disponovat některým z MPD klientů. Pro linuxové distribuce Debian/Ubuntu je to balík *gmpc* (Gnome Music Player Client), tedy grafické rozhraní pro ovládání MPD serveru. Jeho ukázka je viditelná na obrázku 18. Jelikož je na bázi linuxu postaven také stále hojněji užívaný OS Android, existuje i pro tento OS řada MPD klientů. Jako nejstabilnější se ukázal být klient **MP-Droid** který umožňuje jak ovládání hlasitosti a přepínání skladeb, tak i management playlistů a knihovny. GMPC je dostupný také ve verzi pro MS Windows. MPD server dokáže přehrávat nejen zvukové soubory uložené na lokálním, či externím USB disku, ale také například internetová rádia - jak je ostatně patrné i z obrázku 18 kde je právě přehrávaným souborem stream Českého rozhlasu 1 - Radiožurnálu.

6.3.8 Rozšíření o funkci VPN koncentrátoru

Pro Debian existuje pochopitelně mnoho balíků pro nejrůznější způsoby vytvoření VPN, resp. tunelování sítě Internet na druhé, třetí, či vyšších vrstvách modelu RM OSI. Miniserver HomeRouter využívá dvou balíků pro vytvoření VPN koncentrátoru a to *pptpd* pro klienty připojované převážně z klientských stanic s OS MS Windows a *openvpn* pro ostatní embedded zařízení, či klientské stanice např. s OS Ubuntu/Debian. Pro VPN na bázi PPTP protokolu je potřeba konfigurovat soubory */etc/pptpd.conf* kde definujeme lokální a vzdálený rozsah IP adres z naší podsítě, */etc/ppp/chap-secrets* pro konfiguraci uživatelských účtů pro autentizační protokol CHAP a */etc/ppp/pptpd-options* pro konfiguraci povolených autentizačních mechanismů, specifikaci DNS serverů, WINS serverů atd. U OpenVPN je pak potřeba nakonfigurovat soubor */etc/openvpn/server.conf* kde udáváme použitý typ šifrování - symetrické/asymetrické, klíč pro šifrování, klíč certifikační autority atd. Příkladové konfigurace všech zmíněných souborů jsou v příloze A.13

7 Závěr

V jednotlivých kapitolách byl rozebrán způsob jak vytvořit a úspěšně provozovat minimalizovanou softwarovou VoIP ústřednu Asterisk PBX s možností rozšíření o další síťové služby a to vše při současné **spotřebě pouhých několika jednotek Watt** (přesná hodnota je závislá na provozovaných službách a připojených perifériích - viz kapitola 5). Stejně tak bylo v kapitole 5 ukázáno, že **návratnost investice** do tohoto řešení Embedded Asterisk PBX (resp. miniserveru HomeRouter) je ve srovnání s provozováním stejného systému na PC s průměrnou HW konfigurací, **kratší než jeden rok**, což hraje v myšlence Green Computing nebývale významnou roli. Sestavíme-li si, či zakoupíme, takovýto Embedded systém, můžeme dle testů v kapitole 6 dostat softwarovou PBX zvládající **50 současných hovorů bez jejich současného záznamu na externí disk, či 20 současných hovorů se záznamem na zmíněný externí disk**. Navíc je dle grafu v obrázku 16 patrné, že zatížení CPU je téměř lineárně závislé na počtu současných hovorů, což přispívá ke snazší predikci a možnostem optimalizace zmiňovaného systému přesně na míru.

Tak jak bylo zmíněno v kapitole 3.2, myšlenka Green Computing hraje při současné ekonomické situaci většiny nejen evropských států, stále významější roli. Úspory se hledají ve všech společenských i veřejných odvětvích, ICT nevyjímaje. Přesto jsou energetické nároky nutné na provoz samotné sítě Internet minimálně na neuvěřitelných **153 TWh**, a velmi pravděpodobně je dnes tato hodnota ještě vyšší. Proto je potřeba hledat cesty jak energii spotřebovanou na provoz síťové infrastruktury ve firmách, či domácnostech co nejvíce snížit. A právě také proto je již zmiňovaný produkt HomeRouter nasazen ve dvou firmách s počtem do deseti zaměstnanců, kde již více než rok úspěšně a bezproblémově slouží. Investice zmiňovaných firem do řešení HomeRouter, je již dnes zaplacená a nyní se jen aktivně podílí na myšlence Green Computing, přičemž ročně ušetří až okolo čtyř tisíc korun.

8 Reference

- [1] VOZŇÁK, Miroslav, *Voice over IP*, Ostrava: VŠB-TU Ostrava, 2009, ISBN 978-80-248-1828-3.
- [2] CESNET, z.s.p.o., *Asterisk PBX*, URL:<<https://sip.cesnet.cz/cs/swahw/asterisk>>, poslední revize: 12.10.2010, [cit.2012-03-26]
- [3] CESNET, z.s.p.o., WIJA, Tomáš, ZUKAL, David, VOZŇÁK, Miroslav *Asterisk a jeho použití, technická zpráva*
URL:<http://www.cesnet.cz/akce/20051115/pr/voz05_asterisk.pdf>, Praha: 30.10.2005, [cit.2012-03-26]
- [4] HINKLE, Mark *Digium Founder Mark Spencer Recounts the History of Open Source Asterisk PBX*, URL:<<http://socializedsoftware.com/2008/03/16/digium-founder-mark-spencer-recounts-the-history-of-open-source-asterisk-pbx>>, poslední revize: 16. Duben 2008, [cit.2012-03-28]
- [5] Wikimedia Foundation Inc., *Mean opinion score*,
URL:<http://en.wikipedia.org/wiki/Mean_opinion_score>, poslední revize: 3. Prosince 2011 16:39, [cit.2012-03-28]
- [6] Cisco Systems, Inc., *Frame Relay*,
URL:<http://docwiki.cisco.com/wiki/Frame_Relay>, poslední revize: 17. Prosinec 2009, [cit.2012-03-29]
- [7] Cisco Systems, Inc., *Digital Subscriber Line*,
URL:<http://docwiki.cisco.com/wiki/Digital_Subscriber_Line>, poslední revize: 17. Prosinec 2009, [cit.2012-03-29]
- [8] Cisco Systems, Inc., *Ethernet*,
URL:<http://docwiki.cisco.com/wiki/Ethernet_Technologies>, poslední revize: 12. Červen 2011, [cit.2012-03-29]
- [9] IEEE, *IEEE Standard for Local and metropolitan area networks, Virtual Bridged Local Area Networks*, USA: 19. Květen 2006, ISBN 0-7381-4877-6,
URL:<<http://www.dcs.gla.ac.uk/~lewis/teaching/802.1Q-2005.pdf>>
- [10] MACHNÍK, Petr, *Širokopásmové sítě, Kvalita služby*, Ostrava: 15. Listopad 2010
- [11] MACHNÍK, Petr, *Širokopásmové sítě, Nástroje pro implementaci QoS*, Ostrava: 29. Listopad 2010
- [12] MALCOLM, Davenport, RUSSELL Bryant, *IP Quality of Service*,
URL:<[URL:https://wiki.asterisk.org/wiki/display/AST/IP+Quality+of+Service](https://wiki.asterisk.org/wiki/display/AST/IP+Quality+of+Service)>, poslední revize: 30. Srpen 2010, [cit.2012-04-03]

-
- [13] VOZŇÁK, Miroslav, *TELEFONNÍ ÚSTŘEDNY ASTERISK, Teorie a praxe IP telefonie - 3. dvoudenní odborný seminář*, URL:<http://www.ip-telefon.cz/archiv/dok_osta/ipt-2008_Telefonni_ustredny_Asterisk.pdf>, Kongresové centrum Hotelu Olšanka, 5. a 6. listopadu 2008, [cit.2012-04-05]
- [14] ČERVENKA, Marek. Root.cz, *Asterisk 1.8 už není jen telefonní ústředna*, URL:<<http://www.root.cz/clanky/asterisk-1-8-uz-neni-jen-telefonni-ustredna/>>, poslední revize: 18. Listopad 2010, [cit.2012-04-05]
- [15] PatentGenius.com, *Pascal Dornier Patents*, URL:<<http://www.patentgenius.com/inventor/DornierPascal.html>>, poslední revize: neuvédno, [cit.2012-04-10]
- [16] PC Engines, *tinyBIOS*, URL:<<http://www.pcengines.ch/pdf/tb13.pdf>>, Verze 1.3c, [cit.2012-04-10]
- [17] PC Engines, *ALIX system boards* URL:<<http://www.pcengines.ch/alix.htm>>, poslední revize: neuvédno, [cit.2012-04-10]
- [18] PC Engines, *tinyBIOS™ - the open source embedded PC firmware solution*, URL:<<http://www.pcengines.ch/tinybios.htm>>, poslední revize: neuvédno, [cit.2012-04-10]
- [19] Wikimedia Foundation Inc., *Geode (processor)* URL:<[http://en.wikipedia.org/wiki/Geode_\(processor\)](http://en.wikipedia.org/wiki/Geode_(processor))>, poslední revize: 25. Ledna 2012, [cit.2012-04-10]
- [20] Advanced Micro Devices (AMD), One AMD Place, *AMD Geode™ LX Processor Family Technical Specifications*, URL:<<http://www.amd.com/us/products/embedded/processors/geode-lx/Pages/geode-lx-processor-family-technical-specifications.aspx>>, poslední revize: neuvédno, [cit.2012-04-10]
- [21] Advanced Micro Devices (AMD), One AMD Place, *AMD Geode™ LX 800 @ 0.9W Processor Benchmark*, URL:<<http://www.amd.com/us/products/embedded/processors/geode-lx/Pages/geode-lx-800-0.9w-processor-benchmark.aspx>>, poslední revize: neuvédno, [cit.2012-04-10]
- [22] VOZŇÁK, Miroslav, ŘEZÁČ Filip, *Bezpečnost v komunikacích*, 1.vyd, Ostrava: VŠB – Technická univerzita Ostrava, 2011, 101 s.
- [23] Dulce Systems, *Disk Benchmarks for Windows*, URL:<http://www.dulcesystems.com/html/win_bench.html>, poslední revize: neuvédno, [cit.2012-04-11]

-
- [24] Synchromesh Computing, *EMBEDDED x86 PROCESSOR PERFORMANCE RATING SYSTEM WHITE PAPER*,
URL:<<http://www.amd.com/pl/Documents/Synchromesh.PPRWhitePaper-052005.pdf>>, poslední revize: neuváděno,
[cit.2012-04-11]
- [25] Centrum výzkumu konkurenční schopnosti české ekonomiky, BASL, Josef, *Přístupy a trendy v inovacích informačních a komunikačních technologií ve společnosti a ekonomice*,
WORKING PAPER č. 19 (prosinec 2010), Vedoucí: prof. Ing. Antonín Slaný, CSc.,
ISSN 1801-4496
- [26] Voyage Linux, *About Voyage Linux*, URL:<<http://linux.voyage.hk/>>, poslední revize: neuváděno, [cit.2012-04-12]
- [27] Voyage Linux, *Getting Started - PXE boot (v0.8.x)*,
URL:<<http://linux.voyage.hk/content/getting-started-pxe-boot-v08x>>, poslední revize: 3. listopad 2011, [cit.2012-04-12]
- [28] Voyage Linux, DANNY W. Adair *Voyage Linux on USB flash drive*,
URL:<http://wiki.voyage.hk/how_do_i_install_voyage_on_a_usb_flash_drive.txt>, poslední revize: 21. květen 2006, [cit.2012-04-12]
- [29] BELLARD, Fabrice, *Qemu*, URL:<<http://wiki.qemu.org/Manual>>, poslední revize: neuváděno, [cit.2012-04-12]
- [30] Voyage Linux, *Getting Started - Live CD (v0.7.x)*,
URL:<<http://linux.voyage.hk/content/getting-started-live-cd-v07x>>, poslední revize: 9. únor 2011, [cit.2012-04-12]
- [31] Voip-Info.org, *Welcome to the VOIP Wiki - a reference guide to all things VOIP*,
URL:<<http://www.voip-info.org/>>, poslední revize: 17. duben 2012,
[cit.2012-04-17]
- [32] ŠINDELÁŘ, Jan, www.zive.cz, *Kolik energie spotřebuje průměrný počítač?*,
URL:<<http://www.zive.cz/clanky/kolik-energie-spotrebuje-prumerny-pocitac/sc-3-a-126074/default.aspx>>, poslední revize: 8. srpen 2005,
[cit.2012-04-17]
- [33] www.wifi.aspa.cz, ALIX, URL:<<http://wifi.aspa.cz/alix-k22073/>>, poslední revize: neuváděno, [cit.2012-04-17]
- [34] www.alfacomp.cz, *Kingston Compact Flash 4GB*,
URL:<<http://www.alfacomp.cz/php/product.php?eid=105140006000000MVC&Search=compact%20flash%20karta&SearchType=1>>, poslední revize: neuváděno,
[cit.2012-04-17]

- [35] KREJČÍK Ondřej, www.linuxsoft.cz, *Linux v příkazech - správa procesů*, URL:<http://www.linuxsoft.cz/article.php?id_article=409>, poslední revize: 31. září 2004, [cit.2012-04-20]
- [36] Goossens, Michel, *The L^AT_EX companion*, New York: Addison, 1994.
- [37] L^Ampport, Leslie, *L^AT_EX: a document preparation system: user's guide and reference manual*, New York: Addison-Wesley Pub. Co., 1994.

A Konfigurace a návody

A.1 Postup pro instalaci voayge linux z LiveCD

Na stránkách distributora Voyage Linux [30] je dostupný tento přesný návod pro instalaci, krok za krokem:

```
=====
Installing Voyage Linux to a hard disk from Live CD
=====
```

After booting the Live CD, login as root (password: voyage)

1. Create distribution directory for installation

```
# mkdir /tmp/root
# mount -o loop /live/image/live/filesystem.squashfs /tmp/root
# cd /tmp/root
```

2. Make a mount point for installation disk

```
# mkdir /tmp/cf
```

3. Format target disk device

```
# /usr/local/sbin/format-cf.sh /dev/hda
```

This will create /dev/hda1 ext2 partition on /dev/hda disk device.

** Note that this operation is very dangerous since it will erase your disk!
Make sure what you are doing and must do it right!

4. Start voyage.update installation script

```
# /usr/local/sbin/voyage.update
```

Following the instruction to select /tmp/root as distribution directory, and /tmp/cf as mount point. After the installation complete, simple reboot the board and Voyage will be started!

There are some additional packages installed for PXE and NFS server. After starting Voyage, you can safely remove them by:

```
# remountrw
# apt-get remove syslinux atftpd nfs-kernel-server \
    bzip2 sg3-utils minicom
```

After all, remove the last line in /etc/dnsmasq.more.conf:
conf-file=/etc/dnsmasq.pxe.conf

A.2 Nastavení souboru *sip.conf* pro testování pomocí SIPP

Soubor *sip.conf* je zde záměrně zkrácen, protože účty 1000 až 1999 mají zcela stejnou konfiguraci a liší se jen v proměnné *host* kde je specifikováno ze které IP adresy se uživatel může registrovat. Takto byly účty rozděleny do čtyř skupin po 250-ti, přičemž IP adresy byly v rozsahu 10.0.10.2 - 10.0.10.5/24.

```
[general]

context=inner-calls      ; Vychazi kontext pro prichozi hovory

;realm=nasedomena.cz    ; oblast pro souhrnou autentikaci, je vyžadovan nekterymi UA
;bindport=5060           ; UDP Port na kterem je naslouchano – vychazi je 5060
;srvlookup=yes          ; zapnuti DNS SRV lookups pro odchozi hovory
;useragent=Linksys_PAP2T ; Umoznuje zmenit prezentaci UA

registertimeout=20      ; znovu registruj UA kazdych 20s
dtmfmode = rfc2833     ; umoznuje globalne definovat tonovou volbu
;nat=yes                ; Nastaveni NAT, pokud jsou UA za NATem

canreinvite=no         ; RTP pres asterisk

rtpkeepalive=15        ; interval pro zasilani keepalive pro RTP
;domain=nasedomena.cz ; domaina povolena k registraci

;; priklad nastaveni potrebného pro mod B2BUA k Vasemu SIP poskytovateli
;register => vase_cislo:heslo@sip_poskytovatel.cz:5060/
;[ authentication]
;; tvar user:secret@realm
;auth=vase_cislo:heslo@sip_poskytovatel.cz

[1000]
type=friend
context=inner-calls
secret=1234
host=10.0.10.2
dtmfmode=rfc2833
allow=all

.
.
.

[1999]
type=friend
context=inner-calls
secret=1234
host=10.0.10.5
dtmfmode=rfc2833
allow=all
```

A.3 Soubor *extensions.conf* pro testování pomocí SIPP bez současného nahrávání hovorů

Zde je uvedeno kompletní nastavení souboru *extensions.conf* použité pro testování nástrojem SIPP.

```
[general]
static=yes
writeprotect=no
clearglobalvars=no

[globals]

[inner-calls]
; vytoceni prislusneho ucastnika a nasledne zaveseni
exten => _1XXX,n,Dial(SIP/${EXTEN:0},25,t)
exten => _1XXX,n,Hangup
```

A.4 Soubor *extensions.conf* pro testování pomocí SIPP se současným nahráváním hovorů

Zde je uvedeno kompletní nastavení souboru *extensions.conf* použité pro testování nástrojem SIPP.

```
[general]
static=yes
writeprotect=no
clearglobalvars=no

[globals]

[inner-calls]
; nastaveni formatu souboru, do kterych se budou prichazi hovory zaznamenavat
exten => _1XXX,1,Set(CALLFILENAME=${EXTEN:0}-${STRFTIME(${EPOCH},,%Y%m%d-%H%M%S)})
; cesta kam se maji nahravane soubory uložit
exten => _1XXX,n,Monitor(wav,/home/data/nahravky/${CALLFILENAME},mb)
; vytoceni prislusneho ucastnika a nasledne zaveseni
exten => _1XXX,n,Dial(SIP/${EXTEN:0},25,t)
exten => _1XXX,n,Hangup
```

A.5 Konfigurace *extensions.conf* pro funkci podmíněného nahrávání

```
[apps]
;Nastavit nahravani
exten => *7*,1,Set(DB(Record/voice)=1)
exten => *7*,2,Playback(/home/data/nahravky/nahravani_nastaveno)
exten => *7*,n,Hangup

;Zrusit nahravani
exten => **7,1,NoOp(${DB_DELETE(Record/voice)})
exten => **7,2,Playback(/home/data/nahravky/nahravani_zruseno)
exten => **7,n,Hangup

;Detekovat stav nahravani
exten => *77,1,Set(status=${DB(Record/voice)})
exten => *77,n,GotIff(["${status}" = ""]?5)
exten => *77,n,Playback(/home/data//nahravky/nahravani_zapnuto)
exten => *77,n,Hangup
exten => *77,n,Playback(/home/data//nahravky/nahravani_vypnuto)
exten => *77,n,Hangup
```

A.6 Nastavení souborů *interfaces* a *hostapd.wlan0.conf* pro vytvoření AP

Nastavení souboru */etc/network/interfaces*:

```
auto lo
iface lo inet loopback

## WAN interface -> ISP
auto eth0
iface eth0 inet dhcp

## LAN interfaces - eth1,eth2,wlan0
auto br0
iface br0 inet static
address 192.168.3.1
netmask 255.255.255.192
broadcast 192.168.3.63
bridge_ports eth1 eth2 wlan0
up nat.sh br0 eth0 "192.168.3.0/26"
```

Nastavení souboru */etc/hostapd/hostapd.wlan0.conf*:

```
interface=wlan0
driver=nl80211
logger_syslog=-1
logger_syslog_level=2
logger_stdout=-1
logger_stdout_level=2
debug=4
```

```
#dump_file=/tmp/hostapd.dump
#ctrl_interface=/var/run/hostapd
#ctrl_interface_group=0
channel=6
hw_mode=g
macaddr_acl=0
auth_algs=3
eapol_key_index_workaround=0
eap_server=0
wpa=3
ssid=HomeRouter
wpa_passphrase=voyage-linux
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
eapol_version=1
```

A.7 Nastavení souboru *dnsmasq.conf* pro funkci DHCP serveru

```
# file with nameserver for unspecified names
resolv-file=/etc/resolv.conf
# subnet set to 192.168.3.0/26 in /etc/network/interfaces
dhcp-range=192.168.3.7,192.168.3.60,255.255.255.192,24h
# Static DHCP users
# IPPhone - D-link
dhcp-host=00:13:46:d9:15:c3,192.168.3.3
# IPPhone - Ovislink
dhcp-host=00:4f:65:00:00:99,192.168.3.4
# Pepa-WLAN
dhcp-host=1c:4b:d6:84:f1:01,192.168.3.5
# Pepa-LAN
dhcp-host=48:5b:39:23:e6:b8,192.168.3.6
```

A.8 Nastavení souboru *vsftpd.conf* pro funkci FTP serveru

```
listen=YES
local_enable=YES
write_enable=YES
dirmessage_enable=YES
xferlog_enable=YES
connect_from_port_20=YES
xferlog_file=/var/log/vsftpd.log
xferlog_std_format=YES
idle_session_timeout=600
data_connection_timeout=120
anonymous_enable=YES
local_umask=022
anon_upload_enable=YES
anon_mkdir_write_enable=YES
```

A.9 Nastavení souboru *lighttpd.conf* pro funkci HTTP serveru

```

server.modules = (
    "mod_access",
    "mod_alias",
    "mod_compress",
    "mod_redirect",
#    "mod_rewrite",
)

server.document-root      = "/var/www"
server.upload-dirs        = ( "/var/cache/lighttpd/uploads" )
server.errorlog           = "/var/log/lighttpd/error.log"
server.pid-file           = "/var/run/lighttpd.pid"
server.username           = "www-data"
server.groupname          = "www-data"

index-file.names          = ( "index.html", "index.php",
                             "index.htm", "default.htm",
                             "index.lighttpd.html" )

url.access-deny           = ( "" , ".inc" )

static-file.exclude-extensions = ( ".php", ".pl", ".fcgi" )

include_shell "/usr/share/lighttpd/use-ipv6.pl"

dir-listing.encoding      = "utf-8"
server.dir-listing        = "enable"

compress.cache-dir        = "/var/cache/lighttpd/compress/"
compress.filetype         = ( "application/x-javascript", "text/css", "text/html", "text/plain"
                             )

include_shell "/usr/share/lighttpd/create-mime.assign.pl"
include_shell "/usr/share/lighttpd/include-conf-enabled.pl"
fastcgi.server = ( ".php" => (
    "bin-path" => "/usr/lib/cgi-bin/php",
    "socket" => "/tmp/php.socket",
    "max-procs" => "1",
    )))

```

A.10 Nastavení souboru *smb.conf* pro sdílení v sítích Windows

```

[global]
log file = /var/log/samba/log.%m
passdb backend = tdbsam
local master = yes
os level = 255
preferred master = yes
wins support = yes
wins server = 192.168.3.1

```

```

wins proxy = yes
name resolve order = wins lmhosts hosts bcst
dns proxy = yes
remote browse sync
remote announce = 192.168.3.63/DOMA 192.168.6.255/DOMA
writeable = yes
workgroup = DOMA
security = share
syslog = 5
unix charset = UTF-8
panic action = /usr/share/samba/panic-action %d
max log size = 1000
encrypt passwords = true
passwd program = /usr/bin/passwd
domain master = yes
[data]
comment = data
path = /home/data
public = yes
writeable = yes
read only = no
create mode = 0777
directory mode = 0777
[flash]
comment = flash
path = /mnt/
public = yes
writeable = yes
read only = no
create mode = 0777
directory mode = 0777

```

A.11 Nastavení souboru *config.xml* pro sdílení multimédií pomocí UPnP/DLNA

```

<?xml version="1.0" encoding="UTF-8"?>
<config version="1" xmlns="http://mediatomb.cc/config/1" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xsi:schemaLocation="http://mediatomb.cc/config/1 http:$
<server>
  <ui enabled="yes">
    <accounts enabled="no" session-timeout="30">
      <account user="mediatomb" password="mediatomb"/>
    </accounts>
  </ui>
  <ip>192.168.3.1</ip>
  <name>HomeRouter</name>
  <udn>uuid:91e65728-adbc-4302-b8d1-51bb5630eed4</udn>
  <home>/home/data</home>
  <webroot>/usr/share/mediatomb/web</webroot>
  <storage>
    <sqlite3 enabled="yes">
      <database-file>sqlite3.db</database-file>

```

```

</sqlite3>
<mysql enabled="no">
  <host>localhost</host>
  <username>mediatomb</username>
  <database>mediatomb</database>
</mysql>
</storage>
<protocollInfo extend="yes"/><!-- For PS3 support change to "yes" -->
<!--
  Uncomment the lines below to get rid of jerky avi playback on the
  DSM320 or to enable subtitles support on the DSM units
-->
<!--
<custom-http-headers>
  <add header="X-User-Agent: redsonic"/>
</custom-http-headers>

<manufacturerURL>redsonic.com</manufacturerURL>
<modelName>105</modelName>
-->
<!-- Uncomment the line below if you have a Telegent TG100 -->
<!--
  <upnp-string-limit>101</upnp-string-limit>
-->
<!-- ADDED FOR SAMSUNG TV SUPPORT -->
  <custom-http-headers>
    <add header="transferMode.dlna.org: Streaming"/>
    <add header="contentFeatures.dlna.org: DLNA.ORG_OP=01;DLNA.ORG_CI=0;DLNA.
      ORG_FLAGS=01700000000000000000000000000000"/>
  </custom-http-headers>
</server>
<import hidden-files="no">
  <scripting script-charset="UTF-8">
    <common-script>/usr/share/mediatomb/js/common.js</common-script>
    <playlist-script>/usr/share/mediatomb/js/playlists.js</playlist-script>
    <virtual-layout type="builtin">
      <import-script>/usr/share/mediatomb/js/import.js</import-script>
    </virtual-layout>
  </scripting>
  <mappings>
    <extension-mimetype ignore-unknown="no">
      <map from="mp3" to="audio/mpeg"/>
      <map from="ogg" to="application/ogg"/>
      <map from="asf" to="video/x-ms-asf"/>
      <map from="asx" to="video/x-ms-asf"/>
      <map from="wma" to="audio/x-ms-wma"/>
      <map from="wax" to="audio/x-ms-wax"/>
      <map from="wmv" to="video/x-ms-wmv"/>
      <map from="wvx" to="video/x-ms-wvx"/>
      <map from="wm" to="video/x-ms-wm"/>
      <map from="wmx" to="video/x-ms-wmx"/>
      <map from="m3u" to="audio/x-mpegurl"/>
      <map from="pls" to="audio/x-scpls"/>
    </extension-mimetype>
  </mappings>

```

```

    <map from="flv" to="video/x-flv"/>
    <map from="mpg" to="video/mpeg2"/>
    <!-- Uncomment the line below for PS3 divx support -->
    <!-- <map from="avi" to="video/divx"/> -->
    <!-- Uncomment the line below for D-Link DSM / ZyXEL DMA-1000 -->
    <!-- <map from="avi" to="video/avi"/> -->
  </extension-mimetype>
  <mimetype-upnpclass>
    <map from="audio/*" to="object.item.audioItem.musicTrack"/>
    <map from="video/*" to="object.item.videoItem"/>
    <map from="image/*" to="object.item.imageItem"/>
  </mimetype-upnpclass>
  <mimetype-contenttype>
    <treat mimetype="audio/mpeg" as="mp3"/>
    <treat mimetype="application/ogg" as="ogg"/>
    <treat mimetype="audio/x-flac" as="flac"/>
    <treat mimetype="image/jpeg" as="jpg"/>
    <treat mimetype="audio/x-mpegurl" as="playlist"/>
    <treat mimetype="audio/x-scpls" as="playlist"/>
    <treat mimetype="audio/x-wav" as="pcm"/>
    <treat mimetype="video/x-msvideo" as="avi"/>
  </mimetype-contenttype>
</mappings>
</import>
<transcoding enabled="no">
  <mimetype-profile-mappings>
    <transcode mimetype="video/x-flv" using="vlcmpeg"/>
    <transcode mimetype="application/ogg" using="vlcmpeg"/>
    <transcode mimetype="application/ogg" using="oggflac2raw"/>
    <transcode mimetype="audio/x-flac" using="oggflac2raw"/>
  </mimetype-profile-mappings>
  <profiles>
    <profile name="oggflac2raw" enabled="no" type="external">
      <mimetype>audio/L16</mimetype>
      <accept-url>no</accept-url>
      <first-resource>yes</first-resource>
      <accept-ogg-theora>no</accept-ogg-theora>
      <agent command="ogg123" arguments="-d raw -f %out %in"/>
      <buffer size="1048576" chunk-size="131072" fill-size="262144"/>
    </profile>
    <profile name="vlcmpeg" enabled="no" type="external">
      <mimetype>video/mpeg</mimetype>
      <accept-url>yes</accept-url>
      <first-resource>yes</first-resource>
      <accept-ogg-theora>yes</accept-ogg-theora>
      <agent command="vlc" arguments="-I dummy %in --sout #transcode{ venc=ffmpeg, vcodec
        =mp2v, vb=4096, fps=25, aenc=ffmpeg, acodec=mpga, ab=192, samplerate=44100, channel
        $
        <buffer size="1440000" chunk-size="512000" fill-size="120000"/>
    </profile>
  </profiles>
</transcoding>
</config>

```

A.12 Nastavení souboru *mpd.conf* pro funkci síťového přehrávače hudby

```

music_directory      "/home/data/MP3"
playlist_directory  "/var/lib/mpd/playlists"
db_file              "/var/lib/mpd/tag_cache"
log_file             "/var/log/mpd/mpd.log"
pid_file             "/var/run/mpd/pid"
state_file           "/var/lib/mpd/state"
user                 "root"
bind_to_address      "192.168.3.1"
port                 "6600"
gapless_mp3_playback "yes"
save_absolute_paths_in_playlists "yes"
metadata_to_use      "artist,album,title,track,name,genre,date,composer,performer,disc"
follow_outside_symlinks "yes"
follow_inside_symlinks "yes"

audio_output {
    type              "alsa"
    name              "My ALSA Device"
    device            "hw:0,0"      # optional
    format            "44100:16:2" # optional
    mixer_device      "default"     # optional
    mixer_control     "PCM"         # optional
    mixer_index       "0"           # optional
}

mixer_type            "software"
filesystem_charset    "UTF-8"
id3v1_encoding        "UTF-8"

```

A.13 Nastavení konfiguračních souborů pro funkci VPN koncentrátoru

Nastavení souboru */etc/ppp/pptpd.conf*:

```

option /etc/ppp/pptpd-options
debug
logwtmp
bcrelay eth0
localip 192.168.3.1
remoteip 192.168.3.30-60

```

Nastavení souboru */etc/ppp/chap-secrets*:

```

# Secrets for authentication using CHAP
# client      server secret          IP addresses
user1        pptpd  passw1          *

```

Nastavení souboru */etc/ppp/pptpd-options*

```
refuse-pap
refuse-chap
refuse-mschap
require-mschap-v2
ms-dns 192.168.3.1
ms-dns 8.8.8.8
proxyarp
nodefaultroute
lock
nobsdcomp
name pptpd
```

Nastavení souboru */etc/openvpn/server.conf*:

```
dev tun
ifconfig 192.168.2.1 192.168.2.2
tls-server
dh dh1024.pem
ca ca.crt
cert server.crt
key server.key
port 5000
comp-lzo
persist-tun
persist-key
verb 3
```
