

Softwarová pobočková ústředna Asterisk

Software Branch Exchange Asterisk

Bc. Tomáš Knot

Diplomová práce
2013



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Tomáš Knot**
Osobní číslo: **A11438**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Počítačové a komunikační systémy**
Forma studia: **prezenční**

Téma práce: **Softwarová pobočková ústředna Asterisk**

Zásady pro vypracování:

1. Vypracujte literární rešerši na dané téma.
2. Popište vlastnosti a bezpečnost služby.
3. Popište softwarovou pobočkovou ústřednu Asterisk.
4. Nainstalujte a nastavte pobočkovou ústřednu Asterisk.
5. Vytvořte tři laboratorní úlohy, které budou zaměřeny na seznámení s tímto softwarem.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. Asterisk Project: Asterisk Project Wiki [online]. 2012 [cit. 2012-06-26]. Dostupné z: <https://wiki.asterisk.org/wiki/display/AST/Home>
2. Meggelen, J.: Asterisk the Future of Telephony. OReilly, (2006), ISBN 0596009623
3. Vozňák, M.: Voice over IP. Vydavatelství VŠB-TU Ostrava, (2008), ISBN 978-80-248-1828-3
4. Camp, K.: IP Telephony Demystified. McGraw-Hill, (2003), New York, ISBN 0-07-140670-0
5. SOSINSKY, Barrie. Mistrovství – počítačové sítě: Ivše, co potřebujete vědět o správě sítí. Vyd. 1. Brno: Computer Press, 2010, 840 s. ISBN 978-80-251-3363-7.
6. CiscoNetworking Academy: Cisco Systems [online]. 2012 [cit. 2012-06-26]. Dostupné z: <http://www.cisco.com/web/learning/netacad/index.html>

Vedoucí diplomové práce:

prof. Ing. Karel Vlček, CSc.

Ústav počítačových a komunikačních systémů


Datum zadání diplomové práce:

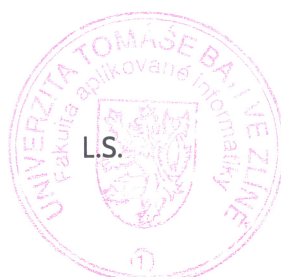
26. února 2013

Termín odevzdání diplomové práce:

31. května 2013

Ve Zlíně dne 26. února 2013


prof. Ing. Vladimír Vašek, CSc.
děkan




prof. Ing. Karel Vlček, CSc.
ředitel ústavu

ABSTRAKT

Cílem diplomové práce je vytvořit softwarovou pobočkovou ústřednu, která bude sloužit v rámci výuky předmětu Telekomunikační systémy. Výsledná ústředna bude vytvořena na platformě operačního systému GNU/Linux Debian a také za pomoci softwaru Asterisk, který zajišťuje propojování telefonních účastníků a ústředen.

Vypracovaná ústředna bude sloužit pro propojování dalších ústředen, které si studenti vytvoří na svých osobních stanicích (forma virtualizace OS Linux a ústředny Asterisk) a bude tedy nadřazenou pobočkovou ústřednou. V rámci diplomové práce se vytvoří modelové příklady, které je možné provádět se studenty v rámci výuky.

Klíčová slova: Linux, Debian, Asterisk, VoIP, IP telefonie, PBX, ústředna

ABSTRACT

The objective of this work is to create a software branch exchange that will be used in the lessons of telecommunications systems. The exchange will be created on the platform of GNU/Linux Debian operating system and with using Asterisk software, which provides the interconnection of telephone subscribers and exchanges.

The exchange will be used for connecting other exchanges that students create on their personal workstations (a form of virtualization Linux OS and PBX Asterisk) and it will be a superior to PBX. This thesis creates model examples that can be performed with students in the classroom.

Keywords: Linux, Debian, Asterisk, VoIP, IP telephony, PBX, exchange

Rád bych poděkoval panu Prof. Ing. Karlu Vlčkovi, CSc. za vedení diplomové práce, za pomoc a rady při psaní této práce. Mé poděkování patří také odbornému konzultantovi Ing. Albertovi Mrázkovi za praktické rady při tvorbě pobočkové ústředny.

Děkuji své rodině za podporu při studiu, kterou mi poskytovala.

„Vědecká práce je naší jedinou cestou k poznání okolní reality.“

Sigmund Freud

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo –bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD	10
I TEORETICKÁ ČÁST	11
1 ZPRACOVÁNÍ ŘEČOVÉHO SIGNÁLU	12
1.1 Co je to řečový signál	12
1.2 PŘEVOD ANALOGOVÉHO SIGNÁLU NA DIGITÁLNÍ A ZPĚT	12
1.2.1 Vzorkování.....	13
1.2.2 Kvantování	14
1.2.3 Rekonstrukce analogového signálu	16
1.3 KÓDOVÁNÍ ŘEČOVÝCH SIGNÁLŮ	16
1.3.1 Kódování tvaru vlny (Wave Coding)	16
1.3.2 Zdrojové kódování (Source Coding)	17
1.3.3 Metody buzení filtru	17
1.3.4 Kodeky v telefonii	18
1.4 HODNOCENÍ KVALITY ŘEČOVÝCH SIGNÁLŮ	19
1.4.1 Měření kvality řečových signálů.....	21
2 ZPŮSOBY KOMUNIKACE V IP TELEFONII	22
2.1 Co je to VoIP (VOICE OVER IP).....	22
2.1.1 Rozdíl mezi klasickou a IP telefoníí.....	23
2.2 VÝVOJ IP TELEFONIE	23
2.3 SIP (SESSION INITIATION PROTOCOL).....	24
2.3.1 Popis protokolu SIP.....	24
2.3.2 Prvky SIPu.....	25
2.3.3 SIP servery a jejich druhy	25
2.3.4 Popis SIP hlavičky.....	26
2.3.5 SIP hlavička žádosti	27
2.3.6 SIP hlavička odpovědi.....	29
2.3.7 Metody SIP	29
2.3.8 Registrace	30
2.3.9 Směrování.....	31
2.4 PŘENOS HLASU PŘES IP	31
3 SOFTWAREVÁ POBOČKOVÁ ÚSTŘEDNA ASTERISK	32
3.1 POUŽITÍ ASTERISKU	32
3.2 PODPORA TECHNOLOGIÍ V ASTERISKU.....	33
3.2.1 Rozhraní	33
3.2.2 Kanály	34

3.2.3	Kodeky	35
3.3	ARCHITEKTURA ASTERISKU	35
3.3.1	Popis architektury	35
3.3.2	Popis modulů API	36
3.4	ČÍSLOVACÍ PLÁN - DIALPLAN	37
3.4.1	Kontexty (Contexts)	37
3.4.2	Klapky (Extensions)	37
3.4.3	Priority (Priorities)	38
3.4.4	Aplikace (Applications)	39
4	FREEPBX	39
5	ZABEZPEČENÍ HLASOVÝCH SLUŽEB	40
5.1	ZABEZPEČENÍ SÍŤOVÝCH PRVKŮ	40
5.1.1	Řízení přístupu k síťovému prvku	40
5.1.2	Oddělení hlasových a nehlasových služeb	41
5.2	ZABEZPEČENÍ SIGNALIZAČNÍHO PROTOKOLU	41
5.2.1	Základní HTTP autentizace	42
5.2.2	Rozšířená HTTP autentizace	42
5.2.3	SIPS (SIP Secure)	42
5.2.4	S/MIME	42
5.2.5	IPsec	43
5.3	ZABEZPEČENÍ TRANSPORTNÍHO PROTOKOLU	43
5.3.1	ZRTP	43
5.3.2	SRTP	44
5.3.3	IPsec	44
II	PROJEKTOVÁ ČÁST	45
6	DŮVODY VYTVOŘENÍ PBX	46
7	INSTALACE OPERAČNÍHO SYSTÉMU GNU/LINUX DEBIAN	46
7.1	Po instalační nastavení	46
7.1.1	Přiřazení uživatele do skupiny sudo	47
7.1.2	Nastavení SSH	48
7.1.3	Přiřazení veřejné IP adresy	48
7.1.4	Zabezpečení firewallu	49
8	INSTALACE PBX ASTERISK	50
8.1	PŘÍPRAVA PŘED INSTALACÍ ASTERISKU	50
8.2	INSTALACE ASTERISKU	51
8.3	INSTALACE FREEPBX	53

9	NASTAVENÍ ÚSTŘEDNY	57
9.1	ZÁKLADNÍ NASTAVENÍ	57
9.1.1	Aktualizace FreePBX.....	58
9.2	VYTVOŘENÍ IAX TRUNKU K POSKYTOVATELI HLASOVÝCH SLUŽEB	58
9.2.1	Vytvoření IAX trunku.....	58
9.2.2	Nastavení odchozí cesty (Outbound Routes)	60
9.2.3	Nastavení příchozí cesty (Inbound Routes)	60
9.3	PŘIDÁNÍ KLAPKY (EXTENSIONS)	61
9.4	INSTALACE PŘÍDAVNÝCH MODULŮ VE FREEPBX	61
9.5	PŘIPOJENÍ STUDENTSKÉ ÚSTŘEDNY KE ŠKOLNÍ ÚSTŘEDNĚ.....	62
9.5.1	Vytvoření IAX trunku na školní ústředně	62
9.5.2	Nastavení odchozí cesty ze školní ústředny	63
9.5.3	Vytvoření IAX trunku na studentské ústředně.....	63
9.5.4	Nastavení odchozí cesty ze studentské ústředny.....	64
10	ZABEZPEČENÍ ÚSTŘEDNY	64
10.1	ZMĚNA VÝCHOZÍCH HESEL.....	64
11	ŘEŠENÍ LABORATORNÍCH ÚLOH	65
11.1	INSTALACE SOFTWAREVÉ POBOČKOVÉ ÚSTŘEDNY ASTERISK	65
11.2	ZABEZPEČENÍ ÚSTŘEDNY.....	66
11.3	NASTAVENÍ ÚSTŘEDNY	66
	ZÁVĚR	67
	ZÁVĚR V ANGLIČTINĚ	69
	SEZNAM POUŽITÉ LITERATURY	71
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	74
	SEZNAM OBRÁZKŮ	76
	SEZNAM TABULEK	78
	SEZNAM PŘÍLOH	79

ÚVOD

Od nepaměti si lidé předávají informace, které jsou důležitým aspektem lidského života. Pomocí informací se člověk dorozumívá se svým okolím. S postupným vývojem a rozvojem techniky byl přenos zrychlován až na úroveň reálného času. Nezapomenutelnou úlohu proto hraje telefonie. Tato technologie umožňuje komunikaci v reálném čase mezi dvěma nebo více lidmi (telekonference). Současný trend telefonie směřuje od klasické PSTN - veřejné telefonní sítě (Public Switched Telephone Network) k IP (Internet Protocol) telefonii, kdy jsou pro přenos hlasových služeb využívány společné datové linky pro síťové i hlasové služby.

Cílem této diplomové práce bude vytvoření VoIP (Voice over IP) pobočkové ústředny založené na softwaru Asterisk, která bude připojena k další nadřazené veřejné SIP (Session Initiation Protocol) ústředně u poskytovatele hlasových služeb. Toto spojení umožní provádět hovory do veřejné telefonní sítě (pevné linky, mobily apod.). Vytvořená ústředna bude sloužit jako centrální bod zprostředkávající komunikaci mezi veřejnou ústřednou a jednotlivými podřízenými pobočkovými ústřednami.

S rozvojem IP telefonie vzrůstá i počet útoků jak na samotný hlasový přenos mezi účastníky hovoru (odposlouchávání, zabránění uskutečnění hovoru atd.), tak také se objevují útoky na ovládnutí pobočkové ústředny (drahá volání apod.). Proto část práce bude věnována zabezpečení komunikace, operačního systému a softwaru Asterisk.

Platforma zabezpečující hlasové služby bude postavena na svobodném operačním systému GNU/Linux Debian. Pro lepší administraci a vizualizaci nastavení ústředny bude nainstalován balík FreePBX. Součástí tohoto balíku je i uživatelská část, která obsahuje obsluhu hlasové schránky a výpisu telefonních hovorů.

V rámci této diplomové práce se vytvoří modelové příklady, s kterými budou studenti pracovat v rámci výuky předmětu Telekomunikační systémy. Hlavní vytvořená ústředna bude sloužit studijním účelům. Ostatní podřízení ústředny budou studenti vytvářet ve výuce a jejich snahou bude propojení a uskutečnění hovoru mezi sebou. Studenti budou mít při práci k dispozici tři laboratorní úlohy. První část je zaměřena na samotné seznámení se softwarem Asterisk (instalace operačního systému GNU/Linux Debian, Asterisk a FreePBX). Druhá úloha je věnována samotnému zabezpečení, jak z pohledu operačního systému, tak také z pohledu ústředny. V poslední úloze se student seznámí s nastavením a fungováním pobočkové ústředny.

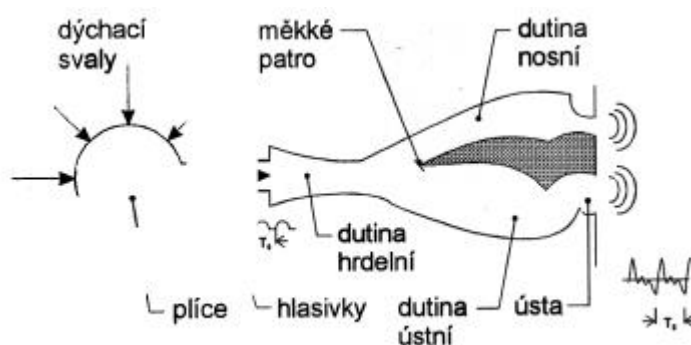
I. TEORETICKÁ ČÁST

1 ZPRACOVÁNÍ ŘEČOVÉHO SIGNÁLU

Při telefonním hovoru je zpracováván hlas účastníka tak, aby jej bylo možné přenést přes digitální přenosové cesty. Tento hlas je označován jako řečový signál. Samotný proces zpracování obsahuje několik kroků, které se musí učinit. Nejprve se provádí snímání mikrofonom a pak přichází na řadu vzorkování a kvantování signálu pomocí A/D (analogově číslicového) převodníku. Poté se kóduje pomocí příslušného kodeku. Na opačné straně účastníka je postup zpracování obrácený. Nejdříve se dekóduje hlas a ten je převeden D/A (číslicově analogovým) převodníkem na akustický zvuk do sluchátka, který slyší účastník.

1.1 Co je to řečový signál

Řečový signál vzniká v řečových orgánech a jsou složeny z hlasivek, hrdelní dutiny, ústní dutiny, nosní dutiny, měkkého patra, tvrdého patra, jazyka a zubů. Plíce společně s dýchacími svaly zajistí buzení řečových orgánů, které proudem vzduchu rozechvějí hlasivky. Zvuk z hlasivek je dále formován dutinou ústní na požadovaný tvar. Dospělý člověk vydává hlas o frekvenci 150 - 400 Hz. Frekvence je odlišná u dítěte, muže nebo ženy. Řečový signál je diskretní, protože má konečnou množinu posloupnosti symbolů, které dokáže jazyk vyjádřit (zhruba 30 - 50). Vydávané hlásky jsou znělé (n, e, atd.) nebo neznělé (č, š, atd.) případně jejich kombinace. Průběh signálu znělých hlásek je kvaziperiodický. U neznělých hlásek má signál průběh podobný šumu. [1, s. 37]



Obr. 1. Řečové ústrojí [1, s. 41]

1.2 Převod analogového signálu na digitální a zpět

Lidské ucho vnímá veškerý zvuk spojitě. Taktéž i hlas je spojitě povahy. Současná telefonní technika je plně digitální a pracuje s digitálním (číslicovým) signálem. Proto je nutné převádět analogový (spojitý) signál na digitální a zpět.

Práce s digitálním signálem přináší výhody, ale má i některé nevýhody. Jednou z velkých nevýhod je, že při digitalizaci dochází k určité ztrátě informace. Velikost ztráty

informace lze ovlivnit změnou parametrů procesu digitalizace. Číslicový signál přináší řadu výhod. Základní výhodou je přenos přes digitální kanály. Oproti analogovým signálům se digitální signály dostávají do místa určení v lepší kvalitě. Je možné detekovat a opravovat chyby (pomocí kódového zabezpečení), které byly způsobeny přenosem. Nedochozí ke zkreslení signálu při přenosu. Dále lze ze signálu vytvářet identické kopie nebo je možné aplikovat šifrovací algoritmy. [2, s. 185 - 186]

- **Analogový (spojitý) signál** - Jedná se o funkci $x(t)$, kde t je čas a je definován pro všechna t v intervalu $(-\infty, +\infty)$. Definiční obor a obor hodnot jsou spojité. Příkladem spojitého signálu může být zvukový signál, který je závislý na čase. [2, s. 186] [3, s. 3]
- **Vzorkovaný signál** - Jedná se o signál, který má diskrétní definiční obor a spojitý obor hodnot. [2, s. 186]
- **Digitální signál** - Jedná se o signál, který má funkci s diskrétním definičním oborem i oborem hodnot (je tvořen posloupností čísel). [2, s. 187]

Digitální signály jsou získávány vzorkováním analogového signálu. A/D převodník provádí převod vzorků na čísla. To znamená, že jsou kvantovány jednotlivé úrovně a jsou vyjádřeny číselným kódem daného převodníku. [4, s. 5]

1.2.1 Vzorkování

Vzorkování je proces, kdy se spojitému signálu v čase $x(t)$ přiřazuje diskrétní signál v čase $x_d(n)$, kde d je index vyjadřující diskrétnost signálu. [4, s. 5]

$$x_d(n) = x(nT) \quad (1)$$

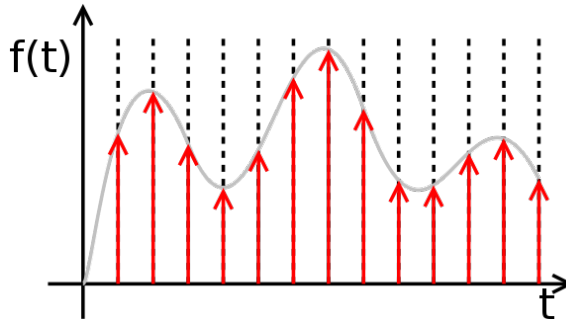
Na celém intervalu se určí význačné body, u kterých se zjistí hodnota. Vzorkovací interval T (vzorkovací perioda) je dán počtem vzorků. Vzorkování je dáno frekvencí, tedy počtem vzorků za vteřinu: [2, s. 187] [4, s. 5]

$$f_{vz} = \frac{1}{T} \quad (2)$$

Je-li vzorkovací frekvence zvyšována, získá se kvalitnější rekonstrukce původního spojitého signálu. [2, s. 187]

Vzorkováním je získán spojitý signál, který je diskrétní v čase. To znamená, že má konečný počet časových okamžiků, ale hodnoty jsou reálná čísla z daného intervalu. Nejčastěji používaný typem je vzorkování s pevnou vzorkovací frekvencí. Dalšími typy vzorkování jsou náhodné a adaptivní. V případě náhodného vzorkování se vzorkovací

body rozdělí náhodně podél časové osy. U adaptivního vzorkování se frekvence vzorkování mění podle určitého kritéria (např. zda dochází ke změně hodnoty signálu). [2, s. 187 - 188] [4, s. 5]



Obr. 2. Vzorkovaný signál [5]

Vzorkovací věta¹⁾ vyslovuje podmínku, při které se vzorkováním signálu neztrácí informace. Její formulace zní: [4, s. 5]

„Pokud signál $x(t)$ spojitý v čase obsahuje pouze frekvenční složky s frekvencemi menšími než f_{max} , pak $x(t)$ může být jednoznačně rekonstruován z posloupnosti ekvidistantních vzorků $x(nT)$, pokud vzorkování frekvence $f_{vz} = 1/T$ je větší než $2f_{max}$, čili“ [4, s. 6]

$$f_{vz} > 2f_{max} \quad (3)$$

Pokud podmínka (3) nemůže být splněna, musí být signál popsán funkcí v čase (analogový signál) frekvenčně omezen, aby této podmínce vyhovoval. [6, s. 6]

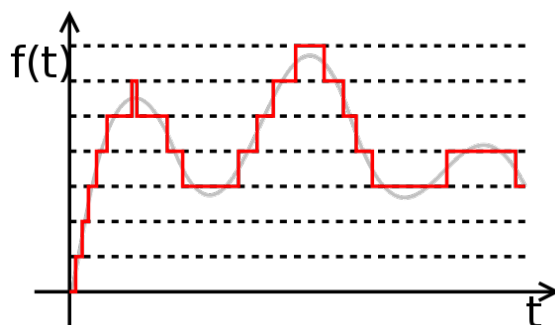
1.2.2 Kvantování

Kvantováním se rozumí převod funkce s diskretním definičním oborem a spojitým oborem hodnot na funkci s diskretním definičním oborem a diskretním oborem hodnot. Převádí se tedy reálné hodnoty na diskretní. [2, s. 192 - 193]

Samotné kvantování je možné chápat jako „zaokrouhlování“. Pokud je zaokrouhlování moc hrubé, je výsledná hodnota kvantování vzdálena originálu. To znamená, že při zpětném převodu bude výsledek více vzdálen původnímu originálu. [2, s. 192 - 193]

Robustní kvantování řeší problém, že u lineárního kvantování dochází se zmenšující se amplitudou ke zmenšování parametru SNR (Signal-to-noise ratio). Snahou je dosáhnout nezávislosti SNR na statistických vlastnostech signálu. Využití robustního kvantování je v PCM (Pulse Code Modulation) telefonu a vyskytuje se ve dvou variantách. První je alaw a druhý je μ law. [6, s. 13 - 16]

¹⁾Autory jsou americký matematik Claude Elwood Shannon a ruský radiotechnik Vladimír Alexandrovič Kotělnikov



Obr. 3. Kvantovaný signál [5]

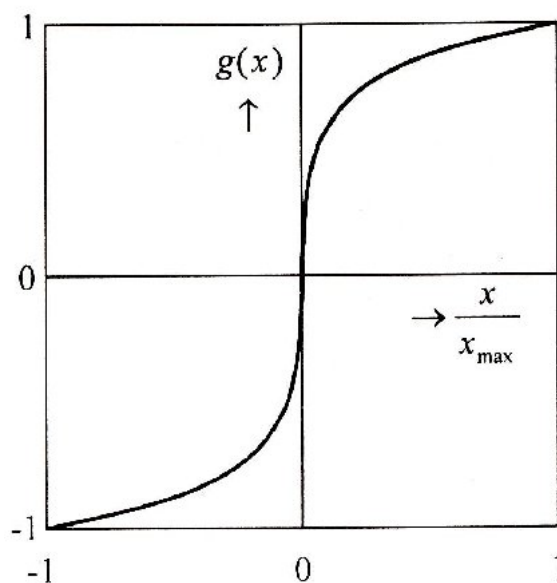
Matematický vztah pro zákon alaw: [6, s. 14]

$$g(x) = \frac{1}{1 + \ln(A)} \ln \left(\frac{x}{x_{max}} \right) + 1 \quad \text{pro } \frac{1}{A} \leq \frac{x}{x_{max}} \leq 1 \quad (4)$$

$$g(x) = \frac{A}{1 + \ln(A)} \frac{x}{x_{max}} \quad \text{pro } 0 \leq \frac{x}{x_{max}} \leq \frac{1}{A} \quad (5)$$

Zákon μ law je definován vztahem: [6, s. 14]

$$g(x) = \frac{\ln \left(1 + \mu \frac{x}{x_{max}} \right)}{\ln(1 + \mu)} \quad \text{pro } x \geq 0 \quad (6)$$



Obr. 4. Kompresní charakteristika pro PCM telefon [6, s. 16]

1.2.3 Rekonstrukce analogového signálu

Rekonstrukcí analogového signálu se myslí zpětný převod digitální podoby do analogové, kterou může vnímat lidský sluch. Rekonstrukce se provádí přes rekonstrukční filtr. Tento filtr je typu dolnofrekvenční (dolní) propusti. Digitální signál je diskrétní posloupností $x[n]$. Není-li dodržena vzorkovací věta, nastává ve frekvenční oblasti k překrývání opakovaných spekter. Frekvenční spektrum je deformováno a není možné získat pomocí rekonstrukce původní signál. Jev, který vzniká, se nazývá aliasing. [2, s. 195] [3, s. 54] [4, s. 9 - 10]

Dalším způsobem je možné provést rekonstrukci pomocí vzorkovače s pamětí, který se v praxi nejčastěji využívá. Digitální signál je D/A převodníkem převeden na spojitý, kdy je držena na výstupu konstantní hodnota napětí až do příchodu dalšího vzorku. [4, s. 11]

1.3 Kódování řečových signálů

Při přenosu řečového signálu digitální cestou musí být dodrženy určité vlastnosti. Tyto vlastnosti se dají shrnout do těchto bodů: co nejmenší počet bitů, co největší kvalita, co nejmenší zpoždění, co největší odolnost proti chybám a co nejmenší výpočetní náročnost. [7, s. 71]

Spojení slov kodéru a dekodéru vzniká zkratka „kodek“ (CODer - DECoder => codec). Cílem je kódovat a dekódovat řečový signál, aby jej bylo možné přenést přes digitální cestu. Kodekem může být software nebo speciální obvod. [8, s. 54]

1.3.1 Kódování tvaru vlny (Wave Coding)

Řečový signál má tvar elektrického signálu. Dochází k zpracování vzorku po vzorku (vzorkování, kvantování a kódování). Tento způsob přináší vysokou kvalitu, ale nese s sebou také velký bitový tok. [1, s. 38] [7, s. 71]

- **Pulzní kódová modulace PCM (Pulse Code Modulation)** - Signál je převáděn do digitální podoby, což má za následek zlepšení odolnosti proti rušení. PCM se skládá ze tří částí - vzorkování, kvantování a kódování (předchozí dva kroky jsou popsány výše). Při kódování jsou diskrétní hodnoty vyjádřeny pomocí n - bitového čísla, kde n je mocnina čísla 2 podle toho, jaké rozlišení je vyžadováno. [1, s. 38] [9, s. 8]
- **Diferenciální pulsně kódová modulace DPCM (Differential Pulse Code Modulation)** - Je modifikací PCM. DPCM je založena na kódování rozdílů mezi okamžitou hodnotou vzorku v určitém vzorkovacím okamžiku a hodnotou, která

je predikována z předchozích vzorků. Redukce přenosové rychlosti a přenášených dat se docílí kódováním rozdílu, kdy je potřeba nižšího počtu bitů [1, s. 40]

- **Adaptivní diferenciální pulsně kódová modulace ADPCM (Adaptive Differential Pulse Code Modulation)** - Vychází ze způsobu kódování DPCM. Zlepšení je dosaženo při srovnávání průběhu, který je vytvářen adaptivně. Tento průběh se přizpůsobuje konkrétním statistickým parametrům řeči. [1, s. 40]

1.3.2 Zdrojové kódování (Source Coding)

Zdrojové kódování patří do parametrických metod, které využívají modelování hlasového traktu, což jsou hlasové syntezátory - tzv. vokodéry. Tímto je dosaženo snížení potřebné rychlosti a získává se lepší komprese signálu při porovnání s kompresí A - zákona u PCM. K realizaci se používá digitálních signálových procesorů a metod zpracování digitálního signálu DSP (Digital Signal Processing). [8, s. 58]

Analogový řečový signál je vzorkován a dále je dělen na segmenty v časových intervalech 10 - 30 ms. Pro dané segmenty se hledají odpovídající parametry zdroje (modelu vokálního traktu a generátoru), které se pak vysílají přes telekomunikační síť. Na straně příjemce se podle přijatých parametrů vytváří řečový signál, jenž se blíží původnímu. Základním parametrem je amplituda, parametr digitálního filtru modelujícího vokální trakt a typ budícího signálu. [8, s. 58]

- **Lineární predikce LP (Linear Prediction)** - Princip je založen na předpovědi následující hodnoty, která je dána výpočtem lineární závislosti z předchozích hodnot. Algoritmy minimalizují střední kvadratickou odchylku mezi predikovanou a skutečnou hodnotou pro 10 - 20 vzorků. Využívá se řešení soustavy rovnic. [1, s. 43] [8, s. 58]
- **Dlouhodobá predikce LPT (Long Term Prediction)** - Princip je založen na korelaci mezi sousedními periodami základního tónu. [8, s. 58]

1.3.3 Metody buzení filtru

Metody buzení filtru patří do parametrických metod a dělí se na: [8, s. 58]

- **Periodické impulsní / šumové buzení** - Je využíváno přepínání mezi impulsním a šumovým generátorem podle toho, jak jsou rozděleny jednotlivé segmenty řeči na znělé a neznělé části. Řečový signál má charakter strojové řeči a je to dáno tím, že obsahuje společně periodickou a šumovou složku. Pouze jedna z těchto složek může převládat. Tohoto principu využívají vokodéry, které nejsou vhodné pro použití v telekomunikacích, kde je potřeba přirozené řeči. [8, s. 58]

- **Buzení zbytkovým signálem RELP (Residually Excited Linear Prediction)** - Používá se zbytkového signálu pro lineární predikci zakódovaného do 1 bitu. [8, s. 58]
- **Multipulzní buzení MPE (Multi - Pulse Excited)** - Je založeno na generování posloupnosti impulsů. Většinou se jedná o 4 - 8 impulsů po 5 ms. Časové polohy a velikosti impulsů se získávají z predikce, jež je důležitá pro digitální filtr, který je optimálně buzen vzhledem k charakteru segmentu řeči. [8, s. 58]
- **Pravidelné pulzní buzení RPE (Regular Pulse Excitation)** - Využívá se toho, že jsou z několika periodických posloupností vybírány jen ty posloupnosti, které jsou vzájemně fázově posunuté. Tím je docíleno snížení výpočetní náročnosti, protože se počítají jen pouze velikosti impulsů. [8, s. 58]
- **Kódové buzení CELP (Code Excited Linear Prediction)** - Používá se blokového vektorového kódování. Snahou je nalezení vzorové posloupnosti v paměti a přenosu její adresy. Tímto je zajištěno zmenšení bitového toku. Pro nalezení posloupnosti se používá vícecestného kódování MSC (Multi - path Search Coding), mezi které patří kódování knihou, stromové a mřížkové. [8, s. 58]

1.3.4 Kodeky v telefonii

Seznam kodeků využívaných v IP telefonii:

- **G.711** - Je to základní kodek, který se používá v PSTN i u VoIP. Převod na digitální signál zajišťuje PCM kódování. Existují dvě charakteristiky vzorkování, které se využívají. Jsou to μ law pro Severní Ameriku, Japonsko a alaw pro zbytek světa. Šířka přenosového pásma je 64 kbit/s. Kvalita hlasu se rovná běžnému telefonnímu hovoru. Kodek má nulové zatížení na CPU (Central Processing Unit) [1, s. 46] [10, s. 194]
- **G.722** - Založeno na ADPCM kódování s přenosovou rychlostí 16 kbit/s. Signál je komprimován v poměru 4:1. Kmitočtové pásmo je rozděleno na 2 části a každé je kódováno samostatně. [1, s. 46]
- **G.723.1** - Využívá dva typy kódování MP - MLQ (Multi - pulse Maximum Likelihood Quantization) s přenosovou rychlostí 6,3 kbit/s nebo ACELP (Algebraic - code - excitation) s přenosovou rychlostí 5,3 kbit/s. [1, s. 46]
- **G.726** - Kodek používá kódování ADPCM a pracuje na několika přenosových rychlostech 16, 24, 32 a 40 kbit/s. PBX (Private Branch Exchange) Asterisk má podporu pro rychlost ADPCM 32 kbit/s. [1, s. 46] [10, s. 194]

- **G.729A** - Je použito kódování CS - ACELP (Conjugate Structure Algebraic Code Excited Linear Prediction). Nabízí výbornou kvalitu hlasu. Přenosová rychlost je 8 kbit/s. Výborný kompresní poměr přináší i zvýšené zatížení CPU. Nevýhodou tohoto kodeku je jeho patentové zatížení. [10, s. 195]
- **GSM** - Komprese kodeku je náročná na výkon CPU. Přenosová rychlost je 13 kbit/s. Kvalita hovoru je o něco menší než u G.729A, ale má výhodu v tom, že není zatížen licenčními poplatky [1, s. 47] [10, s. 195]
- **iLBC (Internet Low Bitrate Code)** - Kodek nabízí vhodnou kombinaci malé šířky přenosového pásma a kvality. Zvláště je pak vhodný pro udržení rozumné kvality při ztrátách na síti. iLBC má podporu v Asterisku, ale není tak populární jako kodeky od ITU (International Telecommunication Union). Kodek nabízí dvě přenosové rychlosti 13,3 kbit/s a 15,2 kbit/s. Vysoká míra komprese je spojena s vysokými nároky na CPU v Asterisku. [10, s. 196]
- **Speex** - Jedná se o kodek, který má variabilní bitrate. To znamená, že se mění šířka přenosového pásma v závislosti na síťových podmínkách. Speex má přenosové rychlosti od 2,15 do 22,4 kbit/s. Velkou výhodou je to, že se jedná o volný kodek, který je vydán pod upravenou variantou licence BSD (Berkeley Software Distribution). [10, s. 196]

Tab. 1. Přehled parametrů kodeků [1, s. 46]

Kodek	Algoritmus	Přenosová rychlost [kbit/s]	MOS
G.711	PCM	64	4,1
G.722	ACELP	16	-
G.723.1	MP - MLQ	6,3	3,9
G.723.1	ACELP	5,3	3,65
G.726	ADPCM	16; 24; 32; 40	3,85
G.729A	CS - ACELP	8	3,7
GSM	RPE - LP	13	3,5
iLBC	-	13,3 nebo 15,2	-
Speex	-	2,15 až 22,4	-

1.4 Hodnocení kvality řečových signálů

Na kvalitě řečového signálu se podílí několik ukazatelů. Samotné hodnocení lze vnímat jako subjektivní a při přenosu je kvalita degradována přenosem, tak i aspektem lidského vnímání. Při posuzování kvality řečového signálu z pohledu lidského vnímání se řadí mezi významné tyto parametry: [11, s. 95]

- **Srozumitelnost** - Z hlediska lidského sluchu se chápe jako věrnost, zřetelnost a nedeformovanost původní signálu. Srozumitelnost hlasu je dána frekvenčním pásmem. Pokud je frekvence vyšší než 1000 Hz, má vliv pouze na dokreslení barvy a charakteru hlasu. [11, s. 95]
- **„End-to-End“ zpoždění** - Jedná se o součet všech zpoždění, které se vyskytnou po celé přenosové cestě a je způsobena vzdáleností společně s prvky na zpracování signálu. Zpoždění nemá přímý vliv na kvalitu řečového signálu, ale ovlivňuje způsob vnímání hovoru. Je-li zpoždění do 100 ms, je většinou nepostřehnutelné. Pokud čas vzroste na 100 - 300 ms, je postřehnutelné „váhání“ v odpovědích účastníků. Zpoždění nad 300 ms jsou patrné a účastníci mají tendenci si vstupovat do rozhovoru, protože se domnívají, že protistrana dohovořila. Komunikace se stává velmi obtížnou. [11, s. 95]
- **Echo** - Je to zvuk, který se vrací zpět ke zdroji. Echo má také označení jako dozvuk či ozvěna. V dnešní době u digitálních systému je echo zvýrazněno u komunikace na krátké vzdálenosti. Způsobují to prvky pro zpracování signálu a další prvky, které se vyskytují po trase. Dosahuje-li echo hodnot do 25 - 30 ms, je vliv na komunikaci minimální. V případě vyšších hodnot je jeho existence nežádoucí. [11, s. 95]
- **Šum** - Přímo ovlivňuje kvalitu řečového signálu, tedy zda je hovor srozumitelný. Šum může vstoupit do hovoru již na začátku jako rušivý zvuk okolí. Dále se objevuje při převodu A/D převodníkem nebo vlivem kodeků, rádiovým přenosem apod. Šum se dostává do popředí v době, kdy se v hovoru objevují místa ticha. [11, s. 96]
- **Nelineární zkreslení** - Použití kodeku se ztrátovou kompresí způsobuje nelineární zkreslení, které je dáno tím, že jsou do řečového signálu vypouštěny redundantní nebo nepodstatné informace. Odstraněním je dosaženo snížení datového toku. [11, s. 96]
- **Ořezání v čase** - Jedná se o způsob, jak pomocí detektoru řečové aktivity VAD (Voice Activity Detector) vyhodnocovat v hovoru pauzu (zda účastník hovoří či nikoliv), kterou není nutné přenášet. Hlas je přenášen pouze jedním směrem, protože většinou hovoří jeden z účastníků. Tímto způsobem se ušetří až 50 % přeneseného objemu dat. Detektor je součástí mobilních telefonů. [11, s. 96]
- **Ztráta paketů** - Ztráty paketů mohou být způsobeny poruchou rádiového kanálu, případně při zaplnění zásobníku směrovače nebo některé z linek jsou neprůchodné. Velké zpoždění paketu a jeho nezařazení do vygenerované řeči může mít

za následek ztrátu paketu, protože informace z něj již nemá relevantní hodnotu. [11, s. 96]

1.4.1 Měření kvality řečových signálů

Optimální metodou pro měření kvality řečových signálů je statistické vyhodnocení názorů dostatečně rozsáhlé skupiny lidí. Existují standardizované testy, které pokrývají tyto způsoby měření. Příkladem je doporučení ITU - T P.82 (Metody pro hodnocení služeb z hlediska kvality přenosu hlasu). Doporučení ITU - T P.830 popisuje způsob výběrů posluchačů, postup příprav řečových vzorků a jejich testování v poslechových testech a vyhodnocení. Je použita pětibodová stupnice MOS (Mean Opinion Score). [11, s. 97]

Tab. 2. Hodnoty pro MOS [7, s. 74]

MOS	Kvalita	Poznámka
1	bad (unacceptable)	velmi rušivý šum a artefakty v signálu
2	poor	...
3	fair	něco mezi
4	good	...
5	excellent	nerozpoznatelné od originálu, bez slyšitelného šumu

Aby bylo hodnocení objektivní, je nutné mít dostatečně velký počet posluchačů pro testování. Každý účastník vnímá hodnocení jinak, protože na něj působí vliv obsahu hovoru apod. Posluchač si většinou vybaví nízkou kvalitu hovoru než hovory, jenž měli dostatečnou kvalitu. Poslechové testy se nahrazují vhodným algoritmem, který zpracovává hlasový vzorek a dále z něj vytvářejí odhad výsledného subjektivního dojmu. [11, s. 97]

Kvalitu řečového signálu je možné měřit dvěma metodami. Jedná se o intrusivní a neinrusivní způsob měření. Intrusivní metoda je založena na měření kvality přenosu hlasu. Je navázáno spojení mezi dvěma stanicemi. Následně je vyslán testovací signál, který je přijat na konci spojení. Poté se porovná původní a přenesený signál daným algoritmem. Poslední část interpretuje výsledky a statisticky je zpracuje. Neinrusivní měření pasivně monitoruje probíhající spojení. Tato metoda je většinou implementována v monitorovacích a dohledových systémech, které monitorují kvalitu přeneseného řečového signálu. Nevýhodou je to, že nemají k dispozici původní signál pro porovnání. [11, s. 98]

Provádění měření kvality řečových signálů se provádí pomocí normalizovaných postupů, které vyžadují určitou skupinu posluchačů, jenž posuzují kvalitu: [7, s. 74]

- **DRT (Diagnostic Rhyme Test)** - Tento způsob se zaměřuje na měření srozu-

mitelnosti podobných párů slov (např. meat x heat) [7, s. 74]

- **DAM (Diagnostic Acceptability Measure)** - Jedná se o soubor několika hodnotících metod, které hodnotí kvalitu komunikačního systému. [7, s. 74]
- **MOS (Mean Opinion Score)** - Je dána skupina 12 - 64 posluchačů, kteří hodnotí kvalitu podle stupnice o pěti bodech. Před samotným hodnocením jsou posluchači „kalibrovaní“ signály, u kterých jsou známy hodnoty MOS. [7, s. 74]

Testy založené na poslechu jsou velmi časově náročné, jak z pohledu organizačního a finančního. Proto jsou vyvinuty dvě techniky, které aproximují výsledky poslechových testů za pomoci technik zpracování signálu. Jsou to doporučení ITU - T P.861 PSQM (Perceptual Speech Quality Measure) a ITU - T P.862 PESQ (Perceptual Evaluation of Speech Quality). [7, s. 74]

2 ZPŮSOBY KOMUNIKACE V IP TELEFONII

Před více než sto lety se začala rozvíjet veřejná telefonie, která byla založena na analogovém principu. Přenos i zařízení pracovaly analogovou formou. S postupným technologickým vývojem se v roce 1972 přechází na PCM, která odstartovala digitalizaci telefonie. V 90. letech 20. stol. se prosazují kodeky G.729, G.723.1 a G.711, jenž se s rozvojem Internetu používají v IP telefonii. [12, s. 38]

V dnešní době dochází ke sblížení informačních a komunikačních technologií. Proto se do popředí dostávají sítě NGN (Next Generation Network). Jedná se filosofický pohled na to, jak by měla vypadat komunikační síť v blízké budoucnosti. V současnosti se NGN rozumí vysokorychlostní digitální sítě, které pracují na principu přepojování okruhů nebo paketů a poskytují telefonní, datové a multimediální služby. Cílem této integrace je spojit systémy přepojování okruhů a směrování do jedné technologie, jenž ve výsledku sníží náklady na pořízení zařízení nutných pro konverzi mezi jednotlivými protokoly a rozhraními s využitím technologie směrovačů IP. [9, s. 8]

2.1 Co je to VoIP (Voice over IP)

VoIP je technologie založena na protokolu IP. Přenos hlasu probíhá za pomoci paketového přenosu (vkládání do paketů). IP telefonie je alternativou ke klasické telefonii, kde je přenos založen na přepojování okruhů přes veřejnou telefonní síť. Přenos veškerých VoIP dat probíhá přes stávající datové sítě, což je jedna z velkých výhod. Není nutné provádět budování nových sítí, ale lze využít stávající infrastrukturu. Tímto jsou sníženy náklady na budování a poskytovatelé nabízejí nižší ceny volání. [12, s. 38] [13, s. 27]

2.1.1 Rozdíl mezi klasickou a IP telefoníí

Jak již bylo zmíněno, základní rozdíl mezi klasickou a IP telefoníí je v typu použité sítě (přepojování okruhů vs. paketový přenos). IP telefonie využívá efektivně jednu přenosovou infrastrukturu, která je společná pro další data. Rozdíl u IP sítě je v tom, že negarantuje odesílateli správné pořadí, úspěšné a včasné doručení adresátovi. Není také udržováno spojení mezi odesílatelem a příjemcem. Při poskytování nových služeb ve sdílené datové síti je nutné dodržet úroveň kvality. Proto je zavedeno řešení kvality služeb QoS (Quality of Service) a má za cíl provádět upřednostňování daných služeb v provozu před ostatními. Každá služba má určitou prioritu. Je-li využita IP telefonie, má pak hlas přednost před ostatním datovým provozem. [12, s. 38] [13, s. 27]

Nevýhodou IP telefonie oproti klasické telefonii je snížení spolehlivosti a dostupnosti služby. Tvrdí se, že se sníží zhruba o 0,5 %. Rozšíření volání přes Internet přineslo řadu nových problémů. Příkladem je např. zneužití VoIP systémů, podvržení cizí identity, spam, zajištění utajení (hovoru i signalizace) a integrity hovoru. [12, s. 38]

2.2 Vývoj IP telefonie

IP telefonie si prošla řadou vývojích etap. V počátcích byla silně orientována na oblast Internetu, jelikož se ke komunikaci používaly IP telefony (softwarové aplikace) a služba nebyla zapojena do veřejné telefonní sítě. [12, s. 39] [13, s. 29]

- **IP - Phone to IP - phone** - Jedná se o počátky IP telefonie, kdy je IP telefon nainstalován jako aplikace na počítači. Cílem bylo odzkoušet možnosti přenosu hlasu po Internetu. [13, s. 29]
- **PBX to PBX** - Patří do druhé fáze vývoje a cílem je propojení vzdálených pobočkových ústředen v korporátní oblasti. Tento model je používán pro interní komunikaci v rámci firmy. [13, s. 29]
- **VoIP to PSTN** - Tento model patří do druhé fáze a je založen na propojení výstupů z IP sítě do veřejné telefonní sítě. Tento model přijala řada telekomunikačních operátorů a nabízejí službu cenově zvýhodněného volání. Nasazuje se také pro zahraniční volání za snížené ceny. [13, s. 29]
- **IP Centrex** - Třetí fází je nabízení metody IP Centrex telekomunikačními operátory. Je to nasazení IP telefonie jako plnohodnotné náhrady klasických pobočkových ústředen a telefonních přípojek. Operátoři nabízejí plnohodnotnou náhradu PBX, kdy jedinou podmínkou je IP konektivita. Páteřní síť musí mít implementován QoS mechanismus pro zajištění kvality nabízených služeb. Analogové přípojky jsou připojeny přes analogové rozhraní, které je zapojeno IP sítě. [13, s. 30]

2.3 SIP (Session Initiation Protocol)

SIP je signalizační protokol, který má na starost signalizaci mezi účastníkem a SIP serverem. Vývoj začal v roce 1996 pracovní skupinou MMUSIC (Multiparty Multimedia Session Control) pod hlavičkou standardizační organizace IETF (Internet Engineering Standard). V roce 2002 byl vydán standard RFC 3216, který popisuje jádro protokolu SIP. Existuje také více než osmdesát RFC, které dále navazují na SIP. [12, s. 91]

2.3.1 Popis protokolu SIP

Protokol SIP pracuje na aplikační vrstvě a používá se pro sestavení, modifikaci a ukončení spojení s jedním nebo více účastníky. Při návrhu bylo dbáno na to, aby protokol byl snadno implementovaný, dostatečně flexibilní a rozšířitelný. Spolu se SIPem se využívají také protokoly RTP (Real - time Transport Protocol) a SDP (Session Description Protocol). RTP se stará o přenos multimédií (hlas, video) v reálném čase. SDP slouží k popisu vlastností účastníků spojení. Přenáší popis, který se používá k vyjednávání parametrů spojení všech zařízení, jenž se účastní komunikace. [12, s. 91]

SIP se řadí k signalizačním protokolům a je end-to-end. Logika je uložena v koncových zařízeních, které si pamatují jednotlivé stavy komunikace. Tento způsob zvyšuje odolnost komunikace proti chybám. Decentralizace a dostupnost přináší zvýšenou režii hlaviček zpráv. Protokol SIP je textově orientovaný. To znamená, že má velmi podobné rysy jako protokoly HTTP (Hypertext Transfer Protocol) a SMTP (Simple Mail Transfer Protocol). Klient zasílá požadavky na server, kde hlavička obsahuje položky From, To nebo Subject. Zde je vidět podobnost s HTTP (zasílání požadavků) a SMTP (položky hlavičky). [12, s. 91]

Entita je u SIPu vázána k doméně obsluhovaného SIP proxy serveru. Mezidoménová komunikace probíhá mezi různými SIP proxy servery. Pouze u multidoménové SIP proxy je výjimka. Entita je v podobě SIP URI (Uniform Resource Identifier). Je to jmenný identifikátor, který má tvar: [12, s. 91 - 92]

```
sip:user:password@host:port;uri-parameters?headers
```

Položka user identifikuje uživatele a host říká, o jakou doménu či hostitele se jedná (kdo má na starost komunikaci s uživatelem). Port je typu UDP (User Datagram Protocol) a má číslo 5060. Parametry se oddělují středníkem. V případě, že je nutné vložit jakékoliv parametry do hlavičky, vkládají se otazníky. Nejčastěji se používá zkrácená verze URI [12, s. 92]

```
sip:user@host
```

SIP překládá jmenné identifikátory URI pomocí DNS (Domain Name System). Příkladem URI může být: [12, s. 92]

```
sip:571123456@sip.fai.utb.cz
```

```
sip:tomas@sip.fai.utb.cz
```

2.3.2 Prvky SIPu

Mezi základní SIP prvky patří UA (User Agents), proxies, registrars a redirects servers. Snahou je všechny logické části SIP serveru (jednotlivé servery) provozovat na společném hardwaru, protože je to efektivní. Koncovými body jsou UA, jenž SIP využívá pro vzájemné spojení. UA je koncový terminál, který může představovat hardwarový SIP telefon nebo aplikaci na počítači, mobilní telefon, PSTN bránu, PDA apod. [12, s. 94] Každý UA obsahuje UAS (User Agent Server) a UAC (User Agent Client). UAS a UAC jsou logickými entitami. UAC vysílá požadavky a přijímá odpovědi. UAS tyto požadavky přijímá a odesílá odpovědi. SIP má dva základní typy zpráv. Je to žádost a odpověď. Koncové zařízení obsahuje UAC i UAS, ale je vždy označeno jako UA. Zvláštním typem UA je B2BUA, které se vkládá do cesty a vytváří dvě spojení. Jedno spojení je ukončeno a nové je sestaveno na cíl. Pro koncové zařízení se chová jako SIP server s rozsáhlými možnostmi, ale výkon je menší než u klasických SIP proxy serverů. Příkladem UAC a UAS může být následující situace. Volající se chová jako UAC. To znamená, že odesílá zprávu INVITE, která je požadavkem na sestavení spojení. Současně také přijímá odpověď na požadavek. Volaný UA pracuje jako UAS. Volaný obdrží zprávu INVITE od volajícího. Změna nastává v případě, že volaný se rozhodne ukončit hovor a odesílá zprávu BYE. Tímto se volající přesouvá do role UAC a volaný do role UAS. [12, s. 94]

2.3.3 SIP servery a jejich druhy

SIP proxy servery tvoří infrastrukturu hostitelů, kteří mají na starost směrování žádostí o spojení podle aktuálního umístění adresáta, provádí autentizaci, účtování hovorného či mohou vykonávat doplňkové služby. Koncová zařízení UA směřují své zprávy na SIP proxy server. Směrování žádostí o sestavení spojení blíže k volanému je nejdůležitějším úkolem pro SIP proxy server. Inicializací spojení jsou prohledávány SIP proxy servery do té doby, až je nalezen nějaký, který zná současné umístění volaného. SIP proxy se snaží přeměřovat žádosti o spojení volajícího k volanému. Ten má možnost žádost o spojení buď přijmout nebo odmítnout. Existují základní dva druhy SIP proxy serverů: [12, s. 95]

- **Stateless (bezstavová) SIP proxy server** - Jedná se o jednoduchý SIP proxy server, který pouze preposílá zprávy bez jejich vzájemných vazeb. Zasiílané zprávy

jsou v pořádku z pohledu souslednosti a významu signalizace. Nekontrolují výměnu z hlediska smysluplnosti, nezachytávají replikaci a detekce nekonečných smyček jim trvá déle. Dále neumí větvení ani přesměrování. Stateless SIP proxy server se používají jako balanční servery, kdy překládají jednoduché zprávy a mají na starost směrování. [12, s. 95]

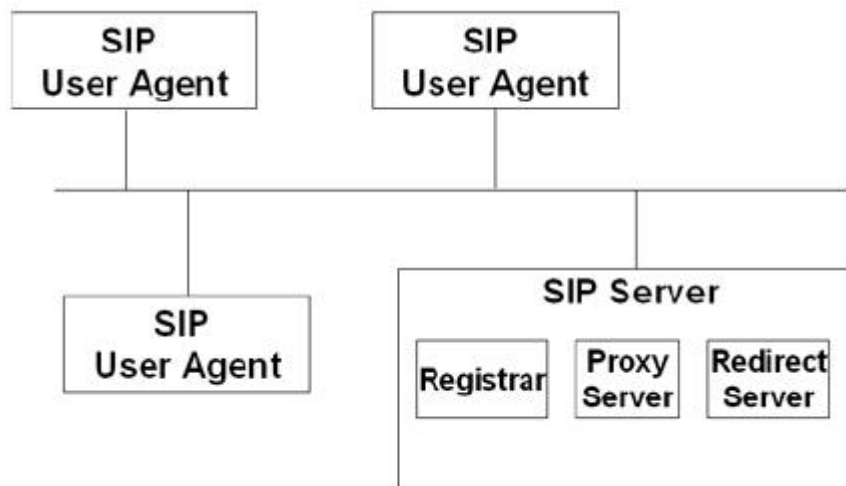
- **Stateful (s informacích o stavech) SIP proxy server** - SIP proxy server je mnohem více komplexnější než stateless. S přijetím požadavku je na serveru vytvořen záznam, který drží stav a důležité záznamy až do ukončení transakce nebo dialogu. Transakční záznam drží stav k žádosti do té doby, dokud není úplně vyřízena. Dialogový záznam uchovává stav dialogu až do ukončení celého spojení. Velká část SIP proxy serverů je právě typu stateful. SIP počítá s možností, že každá jednotka může mít vlastní SIP proxy server, který je používán všemi UA v jedné administrované jednotce. Tyto servery plní úlohu účtování, větvení či některé z nich podporují NAT (Network Address Translation). [12, s. 95 - 96]

Mimo SIP proxy serverů existují i servery: [12, s. 95]

- **Redirect server** - Server má za úkol přijímat požadavky a vyhledávat příjemce, kteří se nacházejí v lokalizační databázi. Databáze je vytvořena Registrar serverem. Poté se vytvoří seznam obsahující aktuální lokalizace uživatele a ten je posílán odesílateli požadavku. Odpovědi jsou zařazeny do třídy 3xx. [12, s. 97]
- **Registrar server** - Jedná se o speciální část serveru, která přijímá požadavky na registraci od uživatelů. Jsou získávány informace o uživateli, což je jejich aktuální poloha (IP adresa, port a uživatelské jméno). Tyto údaje se ukládají do lokalizační databáze (location database). Lokalizační databáze mapuje User URI *sip:tomas@sip.fai.utb.cz* na Device URI *sip:tomas@192.168.1.1:5060*. Jednotlivé registrace jsou limitovány dobou platnosti, která je umístěna v hlavičce kontaktu (položka Expires). V případě, že není obnovena registrace UA, bude daný koncový terminál nedostupný. [12, s. 96 - 97]
- **Location server** - Je to úložiště informací o umístění uživatelů SIP proxy serverů. [12, s. 95]
- **B2BUA** - Je to režim SIP serveru, který se používá. [12, s. 95]

2.3.4 Popis SIP hlavičky

Zprávy v SIPu jsou přenášeny v UDP datagramech. Zpráva má svou hlavičku (header) a vlastní tělo s popisem medií (body, většinou SDP). Hlavička je od těla oddělena pomocí volného řádku CRLF [12, s. 98]



Obr. 5. Model SIPu [14, s. 82]

```

generic-message = start-linek
*header fields
CRLF
[ message-body ]
  
```

První řádek udává, o jaký typ zprávy se jedná. Existují dva druhy zpráv. Je to žádost a odpověď. [12, s. 98]

2.3.5 SIP hlavička žádosti

Žádost slouží k inicializaci procedury (sestavení, ukončení) nebo dává oznámení požadavku příjemci o nějakou událost. Odpověď je potvrzení na žádost, která byla přijata a zpracována. Obsahem je stav zpracování. Ukázka žádosti má následující tvar: [12, s. 98]

```

INVITE sip:102@195.178.94.18:5060 SIP/2.0
Call-ID: 98e0fff0c4dbeca0b91e86035f3f7026@192.168.0.134
CSeq: 310 INVITE
From: "103" <sip:103@195.178.94.18>;tag=2033319271
To: <sip:102@195.178.94.18>
Via: SIP/2.0/UDP 192.168.0.134:51098;
    branch=z9hG4bK5d65cb1abce7dcb18fc7d8bcc3f1045c393538;rport
Max-Forwards: 70
Contact: "103" <sip:103@192.168.0.134:51098;transport=udp>
Content-Type: application/sdp
Authorization: Digest username="103",realm="asterisk",
    nonce="2856b2ed",uri="sip:102@195.178.94.18:5060",
  
```

```
response="ebcfbda6497a55d87dc692939c592917",algorithm=MD5
Content-Length: 299

v=0
o=- 1365421449866 1365421449871 IN IP4 192.168.0.134
s=-
c=IN IP4 192.168.0.134
t=0 0
m=audio 38344 RTP/AVP 96 97 3 0 8 127
a=rtpmap:96 GSM-EFR/8000
a=rtpmap:97 AMR/8000
a=rtpmap:3 GSM/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:127 telephone-event/8000
a=fmtp:127 0-15
```

Z prvního řádku je patrné, že zpráva je INVITE. Slouží k sestavení spojení. URI *sip:102@195.178.94.18:5060* je Request URI a odkazuje na URI dalšího skoku zprávy. Je to next hop a směřuje se podle RURI. Hostitelem je *195.178.94.18* s hledaným uživatelem 102. [12, s. 99]

Dále hlavička obsahuje jednu nebo více položek Via, které slouží k záznamu cesty žádosti. Využita je ke směrování SIP odpovědi stejnou cestou, jakou přišly žádosti. Tato ukázka obsahuje pouze jedno pole Via, které bylo vytvořeno UA odesílatelem. Pole Via udává, že odpověď se má doručit UA na IP adresu 192.168.0.134 na port 51098. [12, s. 99]

Položka From identifikuje volajícího. Pole má ještě parametr tag, který je identifikátorem dialogu. Položka To udává volaného. Hlavička obsahuje i pole Call-ID, jenž je identifikátorem dialogu. Má za úkol identifikovat zprávy, jenž náleží k jednomu volání. Tyto zprávy mají stejný Call-ID. [12, s. 99]

CSeq je pole, jenž čísluje jednotlivé žádosti v dialogu. Přenos zpráv může být občas nespolehlivý, je nutné číslovat, aby příjemce věděl, které žádosti se musí opakovat. Hlavička má pole Contact. Ta obsahuje IP adresu a port. Obě strany si své kontakty vymění a na ně si zasílají další požadavky. [12, s. 99]

Druhá část hlavičky obsahuje popis médií, které vyhovují odesílateli a jsou kódované v SDP. V SDP je popsáno, kdo poslal nabídku SDP a na jaké IP má být ukončen tok médií. Vypsány jsou také kodeky podle preferencí, které se používají. [12, s. 100]

2.3.6 SIP hlavička odpovědi

Odpovědi mají podobnou strukturu jako žádosti. Změna je pouze v prvním řádku, kde je napsána verze protokolu a číslo odpovědi (reply code). Ukázka odpovědi je ve tvaru: [12, s. 101]

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 80.251.250.30;branch=z9hG4bKlycvdkkx;
    received=80.251.250.30;rport=18835
From: "FAI-ip2" <sip:102@195.178.94.18>;tag=uncfq
To: "FAI-ip2" <sip:102@195.178.94.18>;tag=as6664d5b7
Call-ID: biuuwvdmegshffn@thomas-x230
CSeq: 843 REGISTER
Server: FPBX-2.10.1(1.8.20.1)
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER,
    SUBSCRIBE, NOTIFY, INFO, PUBLISH
Supported: replaces, timer
Expires: 3600
Contact: <sip:102@80.251.250.30>;expires=3600
Date: Mon, 08 Apr 2013 14:15:36 GMT
Content-Length: 0
```

Jednotlivé kódy odpovědí jsou celá čísla z rozsahu 100 - 699. Číslo označuje typ odpovědi. Přijetím odpovědi s hodnotou vyšší než 200 je ukončena transakce. Existuje 6 kategorií: [12, s. 101] [15]

- **1xx** - Jsou to takzvané prozatímní odpovědi. Požadavek je přijat a vyzvání. [15]
- **2xx** - Požadavek je úspěšně přijat, správně pochopen a akceptován. [15]
- **3xx** - Zajišťuje přesměrování. Je vytvořen nový požadavek, který je upraven. [15]
- **4xx** - Je to chyba klienta, kdy je vložena špatná syntaxe požadavku. Druhý případ je, že požadavek nemůže být proveden. [15]
- **5xx** - Značí chybu serveru. Server nemůže provést platný požadavek. [15]
- **6xx** - Jedná se o globální chybu. Nelze provést žádný požadavek na serveru. [15]

2.3.7 Metody SIP

Metody (žádosti) slouží k inicializaci procedury. To znamená, že dochází k sestavení, aktualizaci nebo ukončení spojení. RFC 3216 udává šest metod: [12, s. 100]

- **INVITE** - Slouží k zahájení spojení případně ke změně parametrů u probíhajícího spojení (tzv. re - INVITE). [12, s. 100]
- **ACK** - Potvrzuje, že odpověď na žádost INVITE je přijata. Realizace je řešena pomocí „3 - way hand - shaking“. [12, s. 100]
- **BYE** - Metoda, která se používá pro ukončení sestaveného spojení. [12, s. 100]
- **CANCEL** - Volající ruší sestavené spojení, kdy ještě není žádost vyřízena. Proti strana neposlala konečnou odpověď na metodu INVITE. [12, s. 100]
- **REGISTER** - Metoda, která má na starost registraci a rušení registrace. Jmenná adresa uživatele je spojena s IP adresou a portem. Registrace je omezena časem a musí být po určité době obnovována. [12, s. 100]
- **OPTIONS** - Je to speciální typ požadavku, který zjišťuje vlastnosti SIP zařízení. Struktura požadavku je velmi podobná INVITE. Rozdíl je v tom, že není sestaveno spojení. Pouze se přijme odpověď. [12, s. 100]

2.3.8 Registrace

User URI z Form je spojeno s polem Contact z Device URI. IP adresa a číslo portu je zřetelné z Device URI, na kterém lze uživatele zastihnout. Nastane-li případ, že hlavička registrace neobsahuje Contact, vrací se z Registrar serveru, ve formě odpovědi 200 OK seznam platných registrací, které jsou přiřazeny k poli Form z URI. V tomto případě je Device URI *sip:102@80.251.250.30:5060*. V poli Contact je položka expires a udává dobu platnosti registrace. Čas expirace registrace může být měněn Registrar serverem v případě, že UAC nevloží čas, který je v daném intervalu. Je-li nulová hodnota položky expires, jedná se o zrušení registrace. Pro provedení odregistrace je nutné, aby položka expires byla nula a do pole Contact musí být vložen znak *. [12, s. 112 - 113]

Pro zajištění ověření konkrétního uživatele je zavedena registrace s autentizací. Uživatel má přiděleno uživatelské jméno a heslo, které je uloženo v telefonu nebo softwarové aplikaci. Metoda autentizace je stejná jako u HTTP. Používá se metoda HTTP digest authentication. Zabezpečení je založeno na tom, že klient a server mají sdílené tajemství, kterým je heslo. Do jednosměrné funkce se vloží vstupní parametry, jež vytvoří otisk nebo-li hash. Zabezpečení vychází z toho, že není možné získat z otisku vstupní parametry funkce. Hash je jedinečný, a proto nemohou být dva stejné otisky pro různé vstupy funkce. Klient posílá serveru otisk. V případě, že je otisk, který spočetl server stejný, je provedena autentizace. [12, s. 113]

```
REGISTER sip:195.178.94.18 SIP/2.0
```

```
Via: SIP/2.0/UDP 80.251.250.30;rport;branch=z9hG4bKmeqtppdn
```

```
Max-Forwards: 70
To: "FAI-ip2" <sip:102@195.178.94.18>
From: "FAI-ip2" <sip:102@195.178.94.18>;tag=tiyzh
Call-ID: ewevaslzigfjzizy@thomas-x230
CSeq: 810 REGISTER
Contact: <sip:102@80.251.250.30>;expires=3600
Authorization: Digest username="102",realm="asterisk",nonce="3d73d180",
    uri="sip:195.178.94.18",response="b1790dff2f043de3ef6d6
    f4344690c7b",algorithm=MD5
Allow: INVITE,ACK,BYE,CANCEL,OPTIONS,PRACK,REFER,
    NOTIFY,SUBSCRIBE,INFO,MESSAGE
User-Agent: Twinkle/1.4.2
Content-Length: 0
```

2.3.9 Směrování

Směrování v SIPu se využívá DNS nebo statických záznamů a údajů lokalizace UA, které zaslal Registrar server. Snahou je získat z URI adresy cíl. Je-li cíl určen a spojení je navázáno, rozhoduje se, kudy se směřují žádosti. Určuje se, jestli bude SIP proxy prostředníkem při komunikaci a zda bude v signalizační trase SIPu nebo ne. [12, s. 114] Velkou nepříjemností signalizačního protokolu je vytvoření nekonečné smyčky. Dochází k zacyklení signalizačních tras. SIP obsahuje detekci smyček, kdy podle parametru branch v položce Via zjistí, jestli zpráva patří do jedné transakce. Tímto se může limitovat počet zpráv pro danou transakci. Tuto vlastnost mohou využívat pouze stateful SIP proxy servery. Další možností je snižování hodnoty o jedna u pole Max - Forwards v SIP hlavičce. Pokud je hodnota rovna nule, SIP proxy server odešle metodu 485 Too Many Hops a zpráva se ruší. [12, s. 114]

2.4 Přenos hlasu přes IP

Přenos hlasu přes IP se realizuje protokolem RTP (RFC 1889), který patří mezi transportní protokoly. Hlas je vkládán do paketů a je přenášen UDP protokolem. Kontrolu nad přenosem obstarává RTPC (Real - Time Control Protocol, RFC 1890). Jedná se o kontrolní protokol s jehož pomocí, se získávají počty ztracených paketů a proměnné zpoždění (jitter). Existuje rozšíření RTCP XR (Real - Time Control Protocol Extended Reports, RFC 3611), které zjišťuje informace o kvalitě hovoru udávané MOS. RTP přenáší hlas přes porty 10000 - 20000. [13, s. 28] [10, s. 98]

Nastane-li případ, že je nutné komprimovat hlavičku RTP, využije se protokol cRTP (compressed RTP). Kompresí jde hlavičku zmenšit ze 40 oktetů na 2 - 3 oktety. Kom-

prese s sebou nese zvýšené nároky na výkon procesoru směrovačů, jelikož s rostoucím počtem hovorů musí procesor zpracovávat více komprimovaných hlaviček. Před samotným použitím cRTP je potřeba otestovat stabilitu a funkčnost směrovače, zda zvládá přenášet komprimované hlavičky cRTP. [13, s. 28]

3 SOFTWAREVÁ POBOČKOVÁ ÚSTŘEDNA ASTERISK

Asterisk je softwarová pobočková ústředna zvládající IP telefonii, digitální ISDN (Integrated Services Digital Network) a analogovou telefonii. Tedy jedná se o source hybrid TDM (Time Division Multiplex) a packet voice PBX. Ústředna obsahuje také IVR (Interactive Voice Responce), což je automatický odpovídač, který je ovládán přes tónovou volbu DTMF (Dual - tone multi - frequency) či hlasem. Součástí IVR je hlasová schránka pro zanechání vzkazu volajícímu nebo složitější systém pro obsluhu zákaznického účtu (oznámení zůstatku na účtu, informace o účtu apod.). ACD (Automatic Call Distribution) má v ústředně na starost automatické rozdělování hovorů podle daných kritérií (podle určených schémat, podle čísla volajícího, časových podmínek apod.). [16, s. 23] [17, s. 4]

Pobočková ústředna je vydána pod open source licencí GNU GPL (General Public Licence) a běží na platformě GNU/Linux nebo Unix. Nabízené funkce PBX se vyrovnají i komerčním systémům. [16, s. 23] [17, s. 4]

Tab. 3. Hardwarové nároky Asterisku [10, s. 12]

Účel	Počet kanálů	Minimální konfigurace
Domácí systém	Ne více než 5	400 MHz x86, 256 MB RAM
SOHO systém	5 - 10	1 GHz x86, 512 MB RAM
Malý podnikový systém	Až 25	3 GHz x86, 1 GB RAM
Střední až velký podnikový systém	Více než 25	Dvojitě CPU

3.1 Použití Asterisku

Použití Asterisku je možné využít v následujících aplikacích: [16, s. 24] [17, s. 4]

- **PBX i s rozhraním do PSTN**
- **Gateway pro VoIP** - Podporuje protokoly MGCP (Media Gateway Control Protoco), SIP, IAX (Inter - Asterisk eXchange), H.323 [17, s. 4]
- **IVR automatický odpovídač**
- **Hlasová schránka s adresářem**

- **Softwarová ústředna (softswitch)** Je softwarovým řešením komunikačního serveru [16, s. 24] [17, s. 4]
- **Konferenční server** - Obsahuje konferenční místo, funkci Meet me [16, s. 24]
- **Šifrování telefonních a faxových spojení**
- **Překlad čísel**
- **Systém předplaceného volání**
- **Call Center** - centrum volání [16, s. 24]
- **Vzdálené „kanceláře“ pro již existující PBX**
- **Systém pro směrování cestou nejnižších nákladů (LCR)**
- **TDM přes Ethernet**

3.2 Podpora technologií v Asterisku

Asterisk podporuje řadu rozhraní, protokolů a kodeků. Je navržen tak, aby byl snadno rozšířen o další nové technologie, které se objeví v telefonii. [17, s. 5]

3.2.1 Rozhraní

Rozhraní Asterisku se dělí tři základních skupin: [16, s. 27] [17, s. 5]

- **Zaptel (DAHDI²⁾ hardware** - Jedná se o hardware, který zpracovává TDM hardwarově. Dodavatelem těchto pseudo TDM rozhraní je společnost Digium. Rozhraní nabízejí stejnou kvalitu a real - time schopnosti jako hardwarový TDM. Výhodou je nižší cena oproti hardwarovým TDM a větší flexibilita. Rozhraní jsou k dispozici pro různé varianty síťových rozhraní - PSTN, POTS (Plain Old Telephone Service), PRI (Primary Rate Interface), BRI (Basic Rate Interface) a další). V Asterisku je Zaptel hardware nativně podporován. [16, s. 27] [17, s. 5]
- **non - Zaptel hardware** - Jsou to rozhraní, která zajišťují konektivitu u tradičních telefonních služeb. Nejsou podporovány pseudo TDM rozhraní. Mezi rozhraní patří ISDN4Linux, OSS/Alsa, Linux Telephony Interface, Phonejack/Linejack, Dialogic hardware (standardně není podporován). [17, s. 5]
- **Packet voice** - Nejedná se o hardware, ale je to skupina protokolů, která řeší komunikaci přes IP síť. Pro svůj běh nevyžadují specializovaný hardware. Patří sem protokoly SIP, MGCP, IAX / IAX2, H.323, VoFR (Voice over Frame Relay) atd. [16, s. 27] [17, s. 5]

²⁾Od verze Asterisku 1.6 došlo k přejmenování ze Zaptel na DAHDI

3.2.2 Kanály

Pojmem kanál se rozumí logické spojení různých signalizačních a přenosových cest, jenž využívá softwarová pobočková ústředna Asterisk k vytváření a spojování telefonních hovorů. Každý kanál má přiřazeno jedno samostatné volání. Do systému skrze kanál vstupují různé druhy komunikace. Vstupem může být fyzický telefonní okruh (PRI, BRI apod.), softwarová spojení, síťové spojení (SIP, IAX) či vnitřní kanály Asterisku (Agent, Console, Local). Všechny kanály vystupují jako přípojný body. Interakce mezi body se nastavuje v dialplanu, který je v souboru `extension.conf`. Velkou výhodou Asterisku je to, že i v případě, kdy se kanály liší použitou technologií, přistupuje ke všem téměř stejně. Tento způsob zacházení s kanály dělá z ústředny velmi flexibilní platformu. [17, s. 21]

Asterisk podporuje následující typy kanálů:

- **Agent (chan_agent)** - Je to pseudo kanál pro směrování volání ACD agentovi. Nastavení se provádí v konfiguračním souboru `agents.conf`. [17, s. 22]
- **H.323 (chan_h323)** - V Asterisku vystupuje kanál H.323 pouze jako gateway, nikoliv jako gatekeeper. [17, s. 22]
- **IAX2 (chan_iax2)** - Jedná se o protokol, který přenáší signalizaci a hlas. Umožňuje propojení Asterisk serverů a IAX klientů. Konfigurace je uložena v souboru `iax.conf`. [16, s. 27] [17, s. 5, 22]
- **Local (chan_local)** - Je to pseudo kanál, který se využívá k vytvoření smyčky pro volání zpět do dialplanu. [17, s. 23]
- **Modem** - Používá se pouze s ISDN kartou, kterou ovládá ISDN4Linux ovladač. Případně je také možné použít `chan_capi` jako alternativu pro ISDN hardware. [17, s. 23]
- **SIP** - Kanál SIP zabezpečuje komunikaci se SIP telefony a ústřednami. Role Asterisku v rámci SIPu může být SIP klientem, kdy je registrován jako klient k jinému serveru a pouze přijímá, případně umísťuje volání na tento server. Dále lze vystupovat jako SIP server, na který se registrují SIP klienti a Asterisk vytváří spojení mezi ním a klientem. Poslední možností je SIP gateway, jenž je media gateway pro SIP, IAX, MGCP, H.323 či PSTN. Konfigurace se nachází v souboru `sip.conf`. [17, s. 23]
- **MGCP**
- **phone** - Linuxový telefonní kanál [17, s. 22]
- **VoFR**

- **VPB** - Připojení klasického telefonu a telefonní linky, které využívají Voicetronix karty [17, s. 22]

3.2.3 Kodeky

V Asterisku je pouze paketizace 20 ms u protokolů, které používají RTP. Jsou podporovány následující kodeky: [17, s. 6]

- **ADPCM**
- **G.711 μ law, G.711 alaw**
- **G.723.1**
- **G.726**
- **G.729**
- **GSM**
- **iLBC**
- **LPC10**
- **Speex**

3.3 Architektura Asterisku

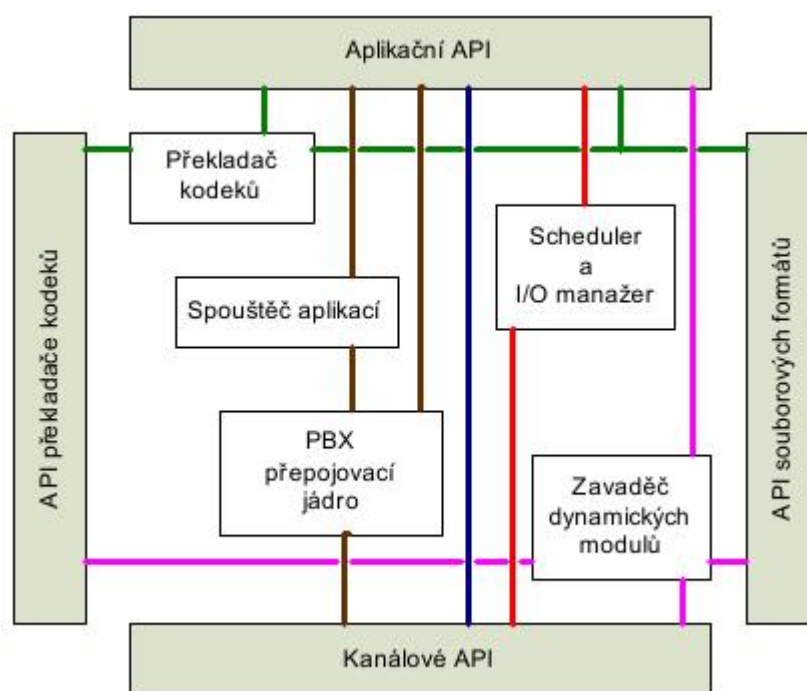
Asterisk má jednoduchou architekturu, čímž se liší od nejčastěji používaných ústředen. Jelikož je vytvořeno stále prostředí, vystupuje ústředna jako středový prvek, který spojuje telefonní aplikace a technologie. Mezi telefonní technologie patří VoIP služby a TDM technologie. Do telefonních aplikací se řadí hlasová pošta, parkování nebo přemostění hovoru, konference atd. Výhodou návrhu je jeho velká flexibilita. [17, s. 6]

3.3.1 Popis architektury

Architektura se skládá z centrálního jádra, okolo něhož jsou definovány specifické API (Application Programming Interface). Jádro ovládá vnitřní propojení ústředny. To znamená, že pracuje s protokoly, kodeky a s hardwarovým rozhraním. Jednotlivé položky, které jsou jádrem ovládány: [16, s. 28] [17, s. 6]

- **PBX přepojování (PBX switching)** - Cílem je provádět spojování volání mezi jednotlivými uživateli a automatizovanými úlohami, která jsou hlavním úkolem PBX. PBX přepojovací jádro propojuje příchozí volání na různých softwarových a hardwarových rozhraních. [16, s. 28] [17, s. 7]

- **Spouštěč aplikací (Application Launcher)** - Zajišťuje spouštění aplikací jako je hlasová pošta, výpis z adresáře, přehrávání souboru atd. [17, s. 7]
- **Překladač kodeků (Codec Translator)** - Využívají se moduly kodeků, jenž kódují a dekódují různé zvukové kompresní formáty. Volbou kodeku lze získat vhodného poměru mezi kvalitou zvuku a šířkou pásma. [17, s. 7]
- **Scheduler a I/O manažer (Schedule and I/O manager)** - Tato část zajišťuje rozvrhování nízkoúrovňových úloh a systémového řízení pro optimální výkon, jenž je ovlivněn stavem zatížení. [17, s. 7]



Obr. 6. Architektura Asterisku [16, s. 28]

3.3.2 Popis modulů API

Zavádění modulů je určeno čtveřicí API a tímto napomáhá k oddělení protokolů a samotného hardwaru. Jelikož se zavádí systém modulů, není nutné řešit detaily ohledně použitých kodeků volajícího nebo jakým způsobem se připojuje apod. [16, s. 29]

- **Kanálové API** - Slouží k ovládání typu spojení příchozího volání (VoIP, PRI, ISDN spojení atd.) Ovládání detailů nižších vrstev spojení mají na starost dynamické jednotky, které jsou zaváděny. [16, s. 29]
- **Aplikační API** - Zabezpečuje chod různých úkolů, pod kterými běží jednotlivé funkce. Je to např. výpis adresáře, hlasová pošta, konference apod. Moduly slouží

systému PBX k ovládání a vykonávání úkolů, které mohou běžet teď nebo v budoucnu. [16, s. 29]

- **API překladač kodeků** - Je to modul pro práci s kodeky a podporou pro různé zvukové formáty (kódování a dekodování). Týká se to kodeků GSM, G.711, MP3 apod. [16, s. 29]
- **API souborových formátů** - Jedná se o modul, který řídí čtení a zápis souborových formátů, jenž ukládá data do souborovém systému. [16, s. 29]

3.4 Číslovací plán - Dialplan

Číslovací plán nebo-li dialpan je srdcem Asterisku. Z tohoto souboru je řízeno směrování příchozích a odchozích hovorů. Skládá se ze seznamu instrukcí nebo kroků, které se vykonávají. Dialplan se nastavuje v souboru extensions.conf a skládá se ze čtyř pojmů: kontextů, klapek, priorit a aplikací. [10, s. 119] [17, s. 8]

3.4.1 Kontexty (Contexts)

Kontexty jsou části dialplanu. Jedná se o pojmenované skupiny klapek, která se využívají k různým účelům. Úlohou kontextu je udržet různé částí dialplanu v interakci s dalšími kontexty. Pokud není nějakým způsobem povoleno, je klapka úplně nezávislá od dalšího kontextu. Kontext je v souboru označen hranatými závorkami []. Pro pojmenování³⁾ se používají velká a malá písmenka včetně číslic. Příkladem pojmenování kontextu může být následující: [10, s. 120] [18]

[odchozi]

Všechny položky instrukcí, které jsou vloženy do kontextu, jsou jeho součástí do té doby, dokud není definován nový kontext. Dialplan má na začátku souboru definované dva kontexty [general] a [globals]. Kontext [general] obsahuje některá obecná nastavení a kontext [globals] definuje globální proměnné v dialplanu. [10, s. 121] [17, s. 8]

3.4.2 Klapky (Extensions)

Klapkou se v telekomunikacích myslí číselný identifikátor pro daný telefon. Asterisk pojem klapka vnímá jako sérii kroků, kde každý krok má přiřazenu aplikaci a přes tuto sérii prochází hovor. Kontext může obsahovat velké množství klapek, které je dáno zaměřením kontextu. Je-li vytočeno číslo dané klapky nebo kanálu, provede pobočková

³⁾Pojmenování kontextu může mít pouze délku pouze 79 znaků. [10, s. 121]

ústředna kroky, které jsou zadány v kontextu. Klapka má určenou syntax a začíná slovem `exten`, kde dále následuje znak rovná se a symbol větší než. Příklad je následující: [10, s. 121 - 122]

```
exten =>
```

Kompletní klapka obsahuje položky jméno (vlozeno může být jméno, číslo nebo kombinace obou), priorita (klapka může obsahovat více kroku, proto se jim přiřazuje priorita) a aplikace (neboli příkaz, který vykoná příslušnou akci na volání). Následující příklad ukazuje seřazení jednotlivých položek v klapce: [10, s. 122]

```
exten => jmeno,priorita,aplikace()
```

Příklad reálné ukázky klapky:

```
exten => 101,1,Answer()
```

3.4.3 Priority (Priorities)

Jak již bylo zmíněno, každá klapka může obsahovat několik kroků. Každá priorita je vyjádřena číslem (začíná se číslem jedna) a vykonává jednu danou aplikaci. [10, s. 122]

```
exten => 101,1,Answer()
```

```
exten => 101,2,Hangup()
```

Z důvodů usnadnění číslování priorit je do Asterisku přidána priorita `n` (má význam další - next). Pokud Asterisk narazí na prioritu `n`, vezme hodnotu předchozího čísla a zvětší ji o jedna. Důležité je, aby vždy první krok měl vloženu prioritu jedna. Jinak klapka nebude k dispozici. Tímto způsobem je jednoduché provádět změny v dialplanu, protože není nutné přepisovat priority u všech kroků. Příklad nečíslované priority je následující: [10, s. 123]

```
exten => 101,1,Answer()
```

```
exten => 101,n,proven nejaky krok
```

```
exten => 101,n,proveden nejaky dalsi krok
```

```
exten => 101,n,Hangup()
```

K nečíslované prioritě lze vložit popisek. Přidání popisku je následující: [10, s. 123]

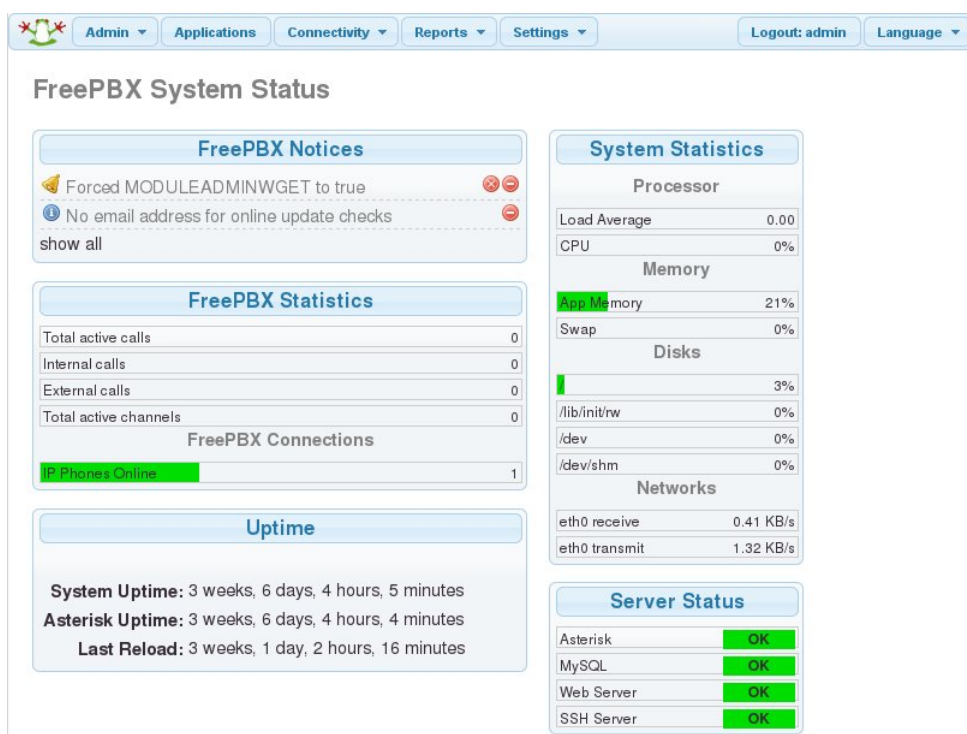
```
exten => 101,n(popisek),aplikace()
```

3.4.4 Aplikace (Applications)

Aplikace jsou důležitou součástí dialplanu. Zajišťují provádění specifických úloh na konkrétním kanálu. Mezi úlohy patří přehrávání zvuku, přijímání tónového vstupu, vytočení kanálu, pokládání hovoru apod. [10, s. 123]

4 FREEPBX

FreePBX je plnohodnotná webová aplikace, která tvoří grafické rozhraní GUI (Graphical User Interface) nad softwarovou pobočkovou ústřednou Asterisk. Ovládání a nastavení ústředny se provádí přes rozhraní webového prohlížeče. Administrace celé ústředny je zjednodušena, protože veškerá nastavení se provádějí skrze tuto aplikaci. Není nutné přistupovat do konfiguračních souborů Asterisk. Veškeré zápisy do těchto souborů provádí FreePBX. FreePBX vyžaduje pro svůj běh webový server Apache, databázový server MySQL a Asterisk. [19] [20] [21]



Obr. 7. Ukázka aplikace FreePBX

Výhodou FreePBX je jeho modularita, protože prostřednictvím repositářů z Internetu lze dohrát jednotlivé moduly, které rozšiřují možnosti využití ústředny. V současné době se vývoj nachází ve verzi FreePBX 2.10. Aplikace je také distribuována jako předpřipravený obraz programu FreePBX s aplikacemi nutnými pro chod a operačního systému (Linux CentOS). V této práci je použita pouze aplikace FreePBX. Operační systém a aplikace jsou nainstalovány samostatně. Není využito předpřipravené distribuce. [19]

FreePBX nabízí po instalaci některé z následujících funkcí: [19]

- **Neomezený počet hlasových schránek**
- **Music on Hold** - Přehrávání MP3 nebo streamováním z Internetu [19]
- **Neomezený počet konferencí** - Omezujícím prvkem je výkon procesoru [19]
- **Fronta hovorů**
- **a další funkce**

5 ZABEZPEČENÍ HLASOVÝCH SLUŽEB

S postupným rozšiřováním se VoIP komunikace stává zajímavým terčem útočníků. Napadení VoIP komunikace je snadnější než v případě klasických telefonních sítí. Pro zabezpečení komunikace je vhodné dodržet určitá pravidla, která sníží rizika.

Jednotlivá pravidla zabezpečení VoIP komunikace lze shrnout do těchto bodů: [22, s. 131]

- **Nastavení řízení přístupu k síťovému prvku na úrovni portů**
- **Oddělení hlasový a nehlasových služeb**
- **Zabezpečení signalizačního protokolu**
- **Zabezpečení transportního protokolu**

Veškeré popisy se týkají protokolu SIP, který je využíván v Asterisku.

5.1 Zabezpečení síťových prvků

Zabezpečení se týká samotných síťových prvků. Jedná se pouze o základní zabezpečení síťových služeb a není závislé na protokolech, které se používají pro hlasové služby.

5.1.1 Řízení přístupu k síťovému prvku

Řízení k síťovému prvku se provádí na úrovni portů, které je definováno doporučením IEEE 802.1x. Doporučení je vydáno institucí IEEE (The Institute of Electrical and Electronics Engineers) a řeší bezpečnou autentizaci v lokálních sítích pomocí bezpečnostních rámců. [22, s. 132]

Bezpečnostní autentizace musí obsahovat následující prvky: [22, s. 132]

- **Suplikant** - Je to prvek, v tomto případě počítač nebo IP telefon, který se bude přihlašovat do sítě. [22, s. 132]
- **Autentizátor** - Jedná se o aktivní síťový prvek, do kterého se připojuje suplikant. Jeho úkolem je být zprostředkovatelem mezi suplikantem a autentizačním serverem. Suplikant spolu s autentizátorem komunikují přes protokol EAPoL (EAP over LAN). Získané údaje od suplikantu předává autentizátor dále autentizačnímu serveru, který rozhoduje o povolení přístupu klienta do sítě. [22, s. 132]
- **Autentizační server** - Úkolem serveru je ověřit identitu suplikantu. Autentizátor a autentizační server mezi sebou komunikují RADIUS protokolem. [22, s. 132]

Zabezpečení přístupu k síťovému prvku je základním bezpečnostním prvkem. Důležité je si uvědomit, že komunikace mezi suplikantem a autentizátorem není šifrována. Data proto mohou být odposlechnuta nebo modifikována. Řešením je využití doporučení IEEE 802.1AE - 2006, které je komplexním bezpečnostním řešením. [22, s. 132]

5.1.2 Oddělení hlasových a nehlasových služeb

Oddělení hlasových a nehlasových služeb se řeší pomocí virtuálních sítí VLAN (Virtual LAN), které je popsáno v doporučení IEEE 802.1q. VVLAN (Voice VLAN) se označují hlasové VLAN. Je doporučeno rozdělit hlasovou a datovou VLAN z důvodu zjednodušení konfigurace sítě a nastavování QoS priorit, protože prioritu lze nastavit pro celou VVLAN. Další výhodou je to, že IP telefony se nemusí zabývat všemi broadcasty v síti a přináší také lepší konfiguraci v oblasti DHCP (Dynamic Host Configuration Protocol) a TFTP (Trivial File Transfer Protocol). [22, s. 132 - 133]

5.2 Zabezpečení signalizačního protokolu

Popis zabezpečení signalizačního protokolu je zaměřeno na protokol SIP, protože jej využívá softwarová pobočková ústředna Asterisk. SIP je svou strukturou návrhu velmi podobný protokolu HTTP. Proto je možné na něj využít bezpečnostní mechanismy, které jsou aplikovány na HTTP. [23, s. 178]

Zabezpečení může být provedeno pomocí protokolu TLS (Transport Layer Security), S/MIME (Secure/Multipurpose Internet Mail Extensions), PGP (Pretty Good Privacy) či IPsec (IP Security) [23, s. 178]

5.2.1 Základní HTTP autentizace

Základní HTTP autentizace přenáší uživatelské jméno a heslo s pomocí záhlaví hlavičky zprávy HTTP - request. Princip funkce umožňuje, aby proxy server nebo koncový UA ověřil identitu klienta či předchozího proxy serveru. Tento způsob autentizace přenáší nešifrované heslo a je možné jej velmi snadno odchytit. Nevýhodou je také to, že se neověřuje integrita dat a data nejsou šifrována. [23, s. 178]

5.2.2 Rozšířená HTTP autentizace

Rozšířená HTTP autentizace je zdokonalením původní základní HTTP autentizace. Funkce je založena na principu výzva - odpověď. Výzva (challenge) obsahuje heslo a náhodný řetězec, která je přivedena na vstup hashovací funkce MD - 5 (Message-Digest - 5) nebo SHA - 1 (Secure Hash Algorithm). [22, s. 133] [23, s. 178 - 179]

Tento způsob autentizace je bezpečnější než v případě základní HTTP autentizace, protože není přenášeno heslo v otevřeném tvaru, ale přenáší se pouze hash. V případě, že je zvoleno velmi slabé heslo, může být hash prolomen slovníkovým útokem. Opět není u této metody kontrola integrity dat a zprávy nejsou šifrovány. [22, s. 133] [23, s. 178 - 179]

5.2.3 SIPS (SIP Secure)

SIPS používá URI ve tvaru *sips:tomas@sip.fai.utb.cz*, kde je přidáno písmeno „s“, které značí zabezpečení. Původní tvar URI je *sip:tomas@sip.fai.utb.cz*. Zpráva INVITE označuje požadavek na zabezpečení celé cesty pomocí protokolu TLS. TLS je lehce inovovaný SSL (Secure Sockets Layer). Každý proxy server na cestě od volaného k volajícímu přidává do SIP hlavičky zprávy vlastní směrovací informace. Zabezpečení TLS je vytvořeno mezi každým zařízením. Podmínkou je použití TCP (Transmission Control Protocol) pro transportní vrstvu a musí existovat PKI (Public Key Infrastructure) pro správu certifikátů. [22, s. 133] [23, s. 179]

5.2.4 S/MIME

S/MIME je zabezpečené MIME. MIME se využívá u emailové komunikace a jeho úkolem je předávat zprávy mezi uživateli. Tělo MIME zprávy může obsahovat text, obrázky, ale i video nebo zvuk. Zabezpečené S/MIME obsahuje kontrolu integrity dat a šifrování. Autentizace veřejného klíče uživatele se využívají X.509 certifikáty. Tělo S/MIME zprávy je zašifrováno symetrickou šifrou a pro dešifrování se používá symetrický klíč. Veřejný klíč příjemce, který se získává z daného certifikátu X.509, musí být ověřen před zašifrováním zprávy. Ověření a získání klíčů probíhá z nějakého veřejného

zdroje či pomocí speciální SIP zprávy. Vydané certifikáty musí být důvěryhodné pro obě strany. Proto je nutné, aby certifikát vydala nějaká certifikační autorita (Thawte, Verisign apod.). Vlastní vydané certifikáty (self - signed) neposkytují dostatečnou míru bezpečnosti. [22, s. 134] [23, s. 179]

5.2.5 IPsec

Jedná se o bezpečnostní mechanismu, který je přidán do standardní IP vrstvy. Vytváří bezpečné šifrované spojení mezi dvěma body. IPsec pracuje na síťové vrstvě a má dva bezpečnostní mechanismy. [22, s. 134] [24]

- **Autentifikace** - Je to mechanismus, který potvrzuje vlastní původ dat. Ověřuje, že přijatý paket je od toho, kdo ho poslal. [24]
- **Kryptování** - Mimo hlavičku IP paketu je vše šifrováno domluvenou šifrou. Odesílatel a příjemce jsou domluveni na způsobu šifrování dat tak, aby příjemce dokázal dešifrovat data. [24]

IPsec je tvořen třemi základními protokoly. Jsou to AH (Authentication Header), ESP (Encapsulating Security Payload) a IKE (Internet Key Exchange). AH má na starost ověření původu paketů. ESP se zaměřuje na ověření dat a šifrování. IKE slouží k výměně klíčů (autentizace pomocí PSK (Pre - shared key) či PKI) a vyjednává parametry spojení [22, s. 134] [24]

5.3 Zabezpečení transportního protokolu

Bezpečnost transportního protokolu je důležitá, jelikož přes něj prochází hlasový stream, který je přenášen protokolem RTP. Transportní protokol pracuje s UDP. Pro zabezpečení se využívá zRTP, SRTP (Secure RTP) nebo IPsec. [22, s. 134] [23, s. 180]

5.3.1 ZRTP

Protokol ZRTP má implementován Diffie - Hellmanova algoritmus pro výměnu klíčů. Tento algoritmus řeší vygenerování tajné sdílené informace, která je pak využita k sestavení zabezpečeného SRTP spojení. ZRTP je rozšířením původního RTP. [22, s. 134] [23, s. 181]

Pro sestavení zabezpečeného spojení není nutné využít PKI nebo PSK klíčů. Jelikož na původní Diffie - Hellmanův algoritmus lze zaútočit útokem typu Man in the Middle, je použit mechanismus SAS (Short Authentication String). Princip SAS je založen na výpočtu hashe dvou Diffie - Hellmanových hodnot. Obě komunikující strany provedou svůj výpočet SAS hodnoty. Skrze jiný komunikační kanál si vymění vypočtené hodnoty.

Pokud jsou hodnoty stejné, pak komunikační kanál s největší pravděpodobností nebyl napaden. [22, s. 134] [23, s. 181] [25]

5.3.2 SRTP

SRTP je rozšířením původního RTP protokolu. Úkolem je zabezpečit transportní kanál. RTP a RTPC pakety mají zajištěnu integritu, autentizaci a ochranu proti replay útokům (útok cílí na opakované přehrávání zpráv). Šifrováno je AES (Advanced Encryption Standard) algoritmem. [22, s. 135] [23, s. 181] [26]

5.3.3 IPsec

IPsec lze také použít pro zabezpečení transportního protokolu. Vlastnosti, které má IPsec u signalizačního protokolu, jsou stejné i pro transportní protokol. Šifruje se algoritmy AES a 3DES (Triple Data Encryption Standard) [22, s. 135] [23, s. 181]

II. PROJEKTOVÁ ČÁST

6 DŮVODY VYTVOŘENÍ PBX

Cílem této diplomové práce je realizace softwarové pobočkové ústředny, která se skládá z počítače, operačního systému GNU/Linux Debian a samotného softwaru Asterisk, jenž je srdcem ústředny. Pro lepší správu je nainstalováno webové rozhraní FreePBX. Součástí práce jsou i laboratorní úlohy, které slouží studentům k seznámení se s principem funkce softwarové pobočkové ústředny, její funkcí a využitím. Výsledná PBX slouží k výuce v rámci předmětu Telekomunikační systémy.

Každý ze studentů má na svém počítači vytvořenou virtualizovanou ústřednu (Linux, Asterisk a FreePBX), kterou budou obsluhovat. Připojením studentské PBX k nadřazené školní ústředně je zajištěna komunikace mezi ostatními studentskými PBX, ale také je umožněn přístup do veřejné telefonní sítě pomocí IAX trunku k VoIP poskytovateli.

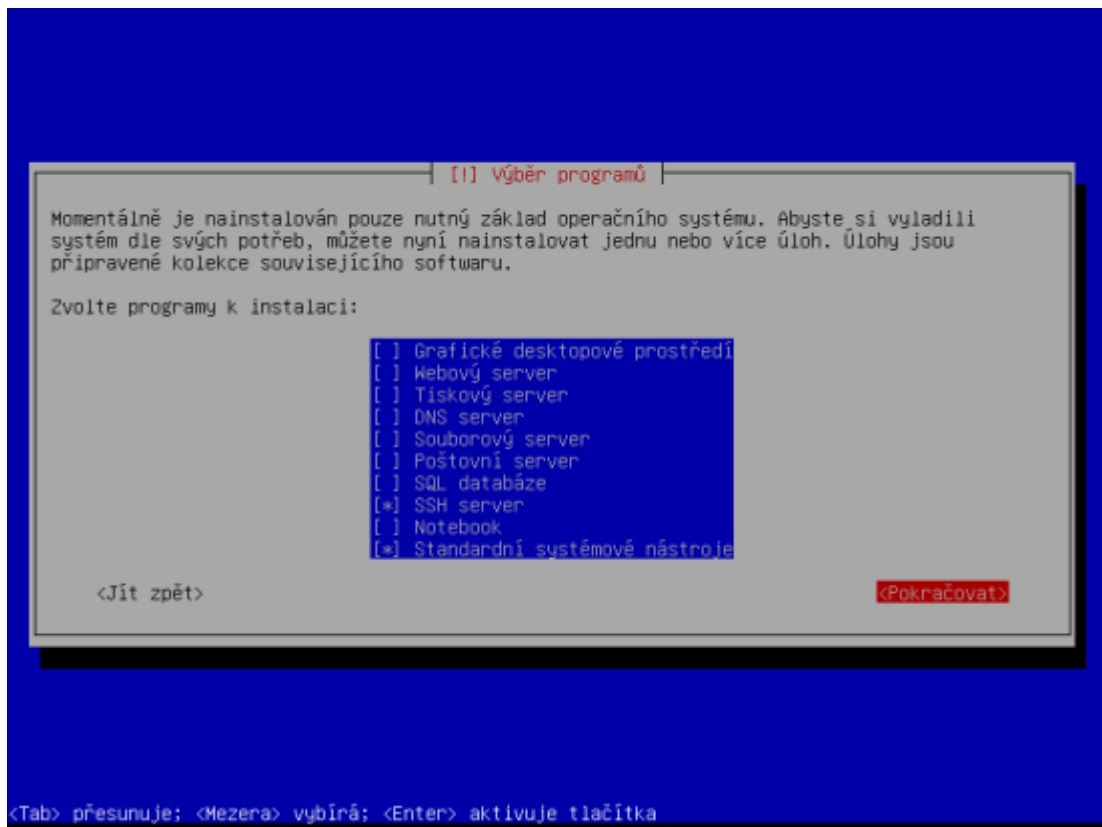
7 INSTALACE OPERAČNÍHO SYSTÉMU GNU/LINUX DEBIAN

Základem celé sestavy PBX je operační systém GNU/Linux Debian Squeeze. Server je před samotnou instalací připojen do školní sítě LAN a má adresu z DHCP. Po nabootování instalačního obrazu z CD, probíhá instalace podle nabízených kroků instalátoru. Jsou vytvořeny dva uživatelé. Prvním je *root* uživatel, který má práva superuživatele. Pro klasickou práci se serverem je také vytvořen uživatel *uziv-tks* s běžnými právy uživatele. Systém je instalován v anglické jazykové mutaci na jeden disk bez RAID (Redundant Array of Inexpensive/Independent Disks) a LVM2 (Logical Volume Management). Veškeré rozdělení disku je ponecháno na systému. Pouze v případě instalace nabídky výběru programů je z instalátoru nainstalován SSH (Secure Shell) server pro vzdálený přístup (Obr. 8).

7.1 Po instalační nastavení

Pro základní důležitá nastavení je využíván uživatel *root*. Veškerá nastavení jsou nutná pro správný a bezpečný chod operačního systému. Nejdříve se provede aktualizace systému pomocí příkazu:

```
# apt-get update && apt-get upgrade
```



Obr. 8. Výběr instalace SSH serveru z instalátoru

7.1.1 Přiřazení uživatele do skupiny sudo

Dalším krokem je přiřazení uživatele *uziv-tks* do skupiny *root*, která má práva pro správu celého systému. Tento uživatel přistupuje k systémovým úkonům a nastavením přes program **sudo**, který pro potvrzení akce vyžaduje heslo uživatele. Výchozí instalace Debianu neobsahuje balíček **sudo** a je nutné jej ručně nainstalovat příkazem:

```
# apt-get install sudo
```

Přiřazení uživatele se provede příkazem:

```
# visudo
```

Otevře se soubor `/etc/sudoers` obsahující nastavení. Do oddílu `# User privilege specification` se vloží následující řádky, které zajistí přístup přes **sudo**:

```
uziv-tks    ALL=(ALL:ALL) ALL
```

7.1.2 Nastavení SSH

Jelikož se k serveru bude přistupovat vzdáleně, je nutné nastavit přístup přes SSH. Konfigurace SSH serveru se nachází v `/etc/ssh/sshd_config`. V souboru `sshd_config` je nutné upravit následující řádky, které zajistí bezpečné připojení.

```
PermitRootLogin no
PasswordAuthentication yes
X11Forwarding no
```

Položka `PermitRootLogin no` zakazuje přihlásit se jako `root` přes SSH. Příkaz, který zajišťuje ověření hesla při přihlášení je `PasswordAuthentication yes`. Poslední je `X11Forwarding no`, kdy se zakáže tunelování X11.

7.1.3 Přiřazení veřejné IP adresy

Pobočková ústředna má přiřazenu veřejnou IP adresu, aby byla přístupná pro připojení k VoIP poskytovateli. Veškeré IP adresy jsou z důvodu bezpečnosti smyšlené.

Nastavení síťového rozhraní se nachází v souboru `/etc/network/interfaces`. Uvnitř souboru se provedou změny. Ve výchozím stavu je nastaveno získání IP adresy z DHCP. Tyto následující položky se ze souboru odstraní nebo se zakomentují, aby nebyly aktivní.

```
auto eth0
allow-hotplug eth0
iface eth0 inet dhcp
```

Místo nich se vloží následující řádky, které přiřadí pevnou IP adresu síťovému rozhraní. Zápis je následující:

```
auto eth0
iface eth0 inet static
address 192.168.1.2
netmask 255.255.255.0
gateway 192.168.1.1
```

Pevná IP adresa 192.168.1.2 je přiřazena síťovému rozhraní `eth0` s maskou 255.255.255.0 a odchozí bránou 192.168.1.1.

7.1.4 Zabezpečení firewallu

Cílem vytvoření firewallového skriptu je zabezpečení celé softwarové pobočkové ústředny. Firewallový skript má na starost filtrování povolených IP adres a služeb, jenž jsou poskytovány serverem. Pravidla skriptu jsou definována pomocí **iptables** a má název **firewall.sh**. Soubor s pravidly je uložen ve složce `/etc/init.d` a spouští se ihned po startu operačního systému.

Skript je rozdělen na dvě části. Část **start_firewall ()** slouží k zavedení nastavených pravidel. Spuštěním této funkce se nastaví všechna pravidla a přístup k serveru mají pouze povolené IP adresy a také jsou povoleny dané služby (porty). Funkce **stop_firewall ()** má na starost smazání všech vložených pravidel a nastavení původních hodnot **iptables**. To znamená, že k serveru má přístup jakákoliv IP adresa a může využít kterýkoliv port.

Firewall má povoleny porty pro služby SSH (TCP 22), SIP (TCP 5060, UDP 5060), DNS (UDP 53), HTTP (TCP 80), RTSP (TCP 554, UDP 554), FTP (TCP 21), FTP-DATA (TCP 20), IXA2 (UDP 4569) a přenos hlasu (UDP 10000 - 20000). Pouze přes tyto porty lze provádět komunikaci s okolím. Pro úspěšnou komunikaci s ústřednou je nutné znát port dané služby, ale také se musí přistupovat z povolené IP adresy.

Jelikož se skript spouští automaticky po startu operačního systému, musí být skript **firewall.sh** umístěn ve složce `/etc/init.d` a také je nutné vytvořit symbolické odkazy na firewallový skript do složek, které reprezentují jednotlivé úrovně běhu systému. Jsou to složky `/etc/rc0.d`, `/etc/rc1.d`, `/etc/rc2.d`, `/etc/rc3.d`, `/etc/rc4.d`, `/etc/rc5.d`, `/etc/rc6.d`. Skriptu se musí přiřadit práva pro spuštění. To se provede následovně:

```
# chmod +x /etc/init.d/firewall.sh
```

Vytvoření symbolických odkazů pro jednotlivé úrovně se provede příkazy:

```
# ln -s /etc/init.d/firewall.sh /etc/rc0.d/K98firewall.sh
# ln -s /etc/init.d/firewall.sh /etc/rc1.d/K98firewall.sh
# ln -s /etc/init.d/firewall.sh /etc/rc2.d/S98firewall.sh
# ln -s /etc/init.d/firewall.sh /etc/rc3.d/S98firewall.sh
# ln -s /etc/init.d/firewall.sh /etc/rc4.d/S98firewall.sh
# ln -s /etc/init.d/firewall.sh /etc/rc5.d/S98firewall.sh
# ln -s /etc/init.d/firewall.sh /etc/rc6.d/K98firewall.sh
```

Při vytvoření symbolických odkazů se vyskytují písmena S a K. Písmeno S označuje spuštění a K ukončení služby. Posledním krokem k automatickému spuštění skriptu je aktualizace seznamu služeb, která se vykoná následujícím příkazem:

```
# update-rc.d firewall.sh defaults
```

Ověření nastavených pravidel se provede příkazem:

```
# iptables -vnL
```

Ovládání firewallového skriptu je také možné manuální cestou. Skript je možné spustit, zastavit nebo restartovat. Spuštění nastaví všechna zadaná pravidla. Zastavení zajistí vymazání všech pravidel a nastavení původních hodnot. Restart ukončí a znovu spustí firewall.

Příkaz pro manuální spuštění:

```
# sh /etc/init.d/firewall.sh start
```

Příkaz pro manuální ukončení:

```
# sh /etc/init.d/firewall.sh stop
```

Příkaz pro manuální restart:

```
# sh /etc/init.d/firewall.sh restart
```

8 INSTALACE PBX ASTERISK

Instalace softwarové pobočkové ústředny se skládá z několika částí. Asterisk není instalován přímo z balíčků, které poskytuje distribuce Debian, ale je instalován ze zdrojových kódu od výrobce a musí být kompilován. Proto se v průběhu instalace stahují balíčky, jež jsou nutné pro kompilaci programu či jeho chod.

Celá instalace je prováděna pod účtem **root**.

8.1 Příprava před instalací Asterisku

Před samotnou instalací Asterisku je potřeba provést stažení linuxového kernelu ve verzi 2.6 a příslušných balíčků, které zajistí kompilaci jádra.

První příkaz stáhne z repositáře zdrojový kód jádra a k němu další důležité programy pro jeho kompilaci. Následuje rozbalení archívu do složky `/usr/src`. Dále se vytvoří symlink. Poté se provede nakopírování konfiguračního souboru a následuje souhrn příkazů pro kompilaci jádra. Postup pro stažení a kompilaci se skládá z těchto příkazů:

```
# aptitude install linux-source-2.6 kernel-package make g++
  libncurses5-dev
# cd /usr/src/
# tar xvjf /usr/src/linux-source-2.6.32.tar.bz2
# ln -s /usr/src/linux-source-2.6.32 /usr/src/linux
# cp /boot/config-`uname -r` /usr/src/linux/.config
# cd linux/
# yes "" | make oldconfig
# make prepare
# make prepare scripts
```

[27]

Dalším nezbytným krokem je instalace encoderu lame a jeho knihovny. Nejprve se musí přidat nový repositář do seznamu pro přidání balíčků.

```
# echo deb http://www.deb-multimedia.org squeeze main non-free >>
  /etc/apt/sources.list
```

[27]

Následně je potřeba provést aktualizaci dostupných balíčků příkazem:

```
# aptitude update
```

Instalace **lame** a **libmp3lame-dev** se provede zadáním:

```
# aptitude install lame libmp3lame-dev
```

[27]

8.2 Instalace Asterisku

Asterisk je instalován ze zdrojových kódů. To znamená, že se musí provést kompilace a poté se nainstaluje. V průběhu instalace probíhá také nastavení parametrů ústředny. Verze Asterisku (duben 2013) je 1.8.21.0.

Prvním krokem je stažení a nainstalování potřebným balíčků, které jsou využity při kompilaci, instalaci a nastavení Asterisku. Získání balíčků se zajistí:

```
# aptitude install libxml2-dev unixodbc-dev libmyodbc subversion  
libmysqlclient15-dev libsqlite0-dev libsnp-dev
```

[27]

V následujícím kroku se provede stažení zdrojový kód Asterisku z Internetu.

Příkazem **wget** je z Internetu stažen aktuální balíček Asterisku, který je poté rozbalen do příslušné složky. Následně se spustí skript **get_mp3_source.sh** a ten stáhne podporu pro soubory MP3. Poté se spustí konfigurace, kdy proběhne kontrola jednotlivých souborů.

Příkaz **make menuconfig** zobrazí menu s výběrem jednotlivých modulů a nastavením kompilace. Z menu se vyberou položky z **Add-ons** → **format_mp3**, **app_mysql**, **app_saycountpl**, **res_config_mysql**, **cdr_mysql**. Dalé se v **Extras Sound Packages** označí položka **EXTRA-SOUNDS-EN-GSM**, což je zvuková sada pro Asterisk. V případě, že se kompilace bude provádět na novější hardwarové sestavě, může docházet při spuštění Asterisku k chybě. Pro správný chod je potřeba v oddíle **Compiler Flags** odznačit **BUILD_NATIVE**. Posloupnost příkazů je:

```
# cd /usr/src/  
# wget http://downloads.asterisk.org/pub/telephony/asterisk/  
old-releases/asterisk-1.8.20.1.tar.gz  
# tar zxvf asterisk-1.8.20.1.tar.gz  
# cd asterisk-1.8.20.1/  
# ./contrib/scripts/get_mp3_source.sh  
# ./configure  
# make menuconfig
```

[27]

V dalším kroku následuje kompilace a instalace Asterisku zadáním příkazů. Příkaz **make** zkompiluje kód a další **make install** provede instalaci Asterisku. Ukázkové nastavení se nainstaluje příkazem **make samples**.

```
# make  
# make install  
# make samples
```

[27]

```
*****
Asterisk Module and Build Option Selection
*****

Press 'h' for help.

---> Add-ons (See README-addons.txt)
Applications
Bridging Modules
Call Detail Recording
Channel Event Logging
Channel Drivers
Codec Translators
Format Interpreters
Dialplan Functions
PBX Modules
Resource Modules
Test Modules
Compiler Flags
Voicemail Build Options
Utilities
AGI Samples
Module Embedding
Core Sound Packages
Music On Hold File Packages
Extras Sound Packages
```

Obr. 9. Nastavení jednotlivých položek před kompilací

8.3 Instalace FreePBX

Instalace grafického rozhraní FreePBX se skládá z několika důležitých částí. Zdrojové kódy FreePBX se získají z Internetu. Dále je nutné nainstalovat i další potřebné balíčky pro chod FreePBX. Verze FreePBX je 2.9.0., která se po instalaci zaktualizuje na nejnovější verzi (duben 2013) 2.10.0. Sled příkazů je:

```
# cd /usr/src/
# wget http://mirror.freepbx.org/freepbx-2.9.0.tar.gz
# tar zxvf freepbx-2.9.0.tar.gz
# aptitude install libxml2 libxml2-dev libtiff4
libtiff4-dev apache2 libapache2-mod-php5 php5-mysql
php5-gd php-pear mysql-server openssl libssl-dev
linux-source-2.6.32 bison libncurses5-dev
libaudiofile-dev curl sox
```

[27]

Při instalaci balíčku **mysql-server** je uživatel vyzván k zadání hesla, které využívá *root* MySQL uživatel.

V konfiguračním souboru `/etc/php5/apache2/php.ini` se provede změna parametrů u položek **upload_max_filesize** (povolení maximální velikost souborů pro upload) a

memory_limit (limit paměti, které může využít skript). Příkazy pro snadnou úpravu jsou následující:

```
# sed -i "s/(upload_max_filesize *= *\\)(.*)/\\120M/"
    /etc/php5/apache2/php.ini
# sed -i "s/(memory_limit *= *\\)(.*)/\\1100M/"
    /etc/php5/apache2/php.ini
```

[27]

Nezbytným krokem je také vytvoření samostatného uživatele a skupiny, který bude sloužit pro chod FreePBX. Uživatel je pojmenován *asterisk* a skupina má opět stejné jméno *asterisk*. Vytvoření skupiny a uživatele se provede příkazy:

```
# groupadd asterisk
# useradd -c "Asterisk PBX" -d /var/lib/asterisk -g
    asterisk asterisk
```

[27]

V následující posloupnost příkazů vytváří databáze (*asteriskcdrdb*, *asterisk*) a uživatele *asteriskuser* s heslem *amp109* na MySQL serveru.

```
# mysql -u root -p
# Enter password:
# mysql> create database asteriskcdrdb;
# mysql> create database asterisk;
# mysql> GRANT ALL PRIVILEGES ON asteriskcdrdb.*
    TO asteriskuser@localhost IDENTIFIED BY 'amp109';
# mysql> GRANT ALL PRIVILEGES ON asterisk.*
    TO asteriskuser@localhost IDENTIFIED BY 'amp109';
# mysql> quit;
```

[27]

Vytvořené databáze pro běh FreePBX jsou zatím prázdné. Vložením příslušných příkazů se naplní databáze *asteriskcdrdb* a *asterisk*. Vkládání údajů do tabulek se provádí pod uživatelem *asteriskuser*.

```
# mysql -u asteriskuser -pamp109 asteriskcdrdb <
  /usr/src/freepbx-2.9.0/SQL/cdr_mysql_table.sql
# mysql -u asteriskuser -pamp109 asterisk <
  /usr/src/freepbx-2.9.0/SQL/newinstall.sql
```

[27]

Následující příkazy se zaměřují již na samotnou instalaci balíku FreePBX. Předchozí kroky se věnují přípravě, která musí být splněna před tím, než se provede instalace FreePBX. Nejdříve se vytvoří záloha souboru `/etc/asterisk/sip_notify.conf`. Příkazem `start_asterisk start` se spustí Asterisk. Je také nutné doinstalovat doplněk pro PHP, který vyžaduje FreePBX pro svůj chod.

```
# mv /etc/asterisk/sip_notify.conf /etc/asterisk/
  sip_notify.conf.backup
# /usr/src/freepbx-2.9.0/start_asterisk start
# pear install DB
# /usr/src/freepbx-2.9.0/install_amp
```

[27]

Spuštěním příkazu `/usr/src/freepbx-2.9.0/install_amp` se provede instalaci FreePBX. Instalátor je interaktivní a uživatele vybízí k zadávání údajů. Během instalace je uživatel vyzván k těmto událostem:

```
Enter your USERNAME to connect to the 'asterisk' database:
  [asteriskuser]
Enter your PASSWORD to connect to the 'asterisk' database:
  [amp109]
Enter the hostname of the 'asterisk' database: [localhost]
Enter a USERNAME to connect to the Asterisk Manager interface:
  [admin]
Enter a PASSWORD to connect to the Asterisk Manager interface:
  [amp111]
Enter the path to use for your AMP web root: [/var/www/html]
  /var/www
Enter the IP ADDRESS or hostname used to access the AMP
  web-admin: [192.168.1.1] 192.168.1.11
Enter a PASSWORD to perform call transfers with the
  Flash Operator Panel: [passw0rd]
Use simple Extensions [extensions] admin or separate Devices and
```

```
Users [deviceanduser]? [extensions]
Enter directory in which to store AMP executable scripts:
[/var/lib/asterisk/bin]
Enter directory in which to store super-user scripts:
[/usr/local/sbin]
```

[27]

Všechny hodnoty jsou nastaveny jako výchozí kromě bodu pro zadání cesty a IP adresy pro přístup k webovému rozhraní.

Běh webového rozhraní má na starost webový server Apache. Pro případ nějaké chyby v konfiguraci se udělá záloha konfiguračního souboru `/etc/apache2/apache2.conf`. Poté je možné provést požadované změny. U Apache se nastaví přístupová práva pro uživatele a skupinu *asterisk* v daném souboru. Z důvodu provedených změn je potřeba zastavit Apache (příkaz **apache2 stop**) a poté ho znovu (příkaz **apache2 start**) spustit. Příkazy pro úpravu přístupových práv jsou následující:

```
# cp /etc/apache2/apache2.conf /etc/apache2/apache2.conf-orig
# sed -i "s/\(^User *\)\(.*\)\/\1asterisk/" /etc/apache2/
    apache2.conf
# sed -i "s/\(^Group *\)\(.*\)\/\1asterisk/" /etc/apache2/
    apache2.conf
# /etc/init.d/apache2 stop
# /etc/init.d/apache2 start
```

[27]

Poslední krokem instalace je vytvoření spouštěcího skriptu, který zajišťuje spuštění **amportalu**. To znamená, že Asterisk se spouští ihned po startu operačního systému. Skript je umístěn ve složce `/etc/init.d` a má název **amportal-startup**. Zdrojový kód je vložen v příloze. Příkaz pro vytvoření souboru otevře textový editor **nano**, do kterého se vloží příslušný zdrojový kód:

```
# nano /etc/init.d/amportal-startup
```

[27]

Vytvořenému souboru je potřeba přiřadit práva pro spuštění. To se provede příkazem:

```
# chmod +x /etc/init.d/amportal-startup
```

[27]

Skript je nutné umístit do tzv. init skriptů, které se spouštějí po startu operačního systému. Následující příkaz zajistí jeho přiřazení:

```
# inserv /etc/init.d/amportal-startup
```

[27]

Nyní se provede restart systému a po jeho načtení je možné přes webový prohlížeč přistoupit ke konfiguraci FreePBX.

```
# reboot
```

9 NASTAVENÍ ÚSTŘEDNY

Instalací ústředny práce nekončí, naopak začíná to nejdůležitější. Správné nastavení PBX je nutné pro její správný a bezproblémový chod. Tato kapitola se věnuje základnímu nastavení ústředny a jejímu zprovoznění.

9.1 Základní nastavení

Veškeré nastavení ústředny se provádí přes webové rozhraní FreePBX. Do webového prohlížeče se vloží IP adresa serveru např. *http://192.168.1.11/admin*. Prozatím je nastaveno výchozí uživatelské jméno a heslo pro administrátora.

Uživatelské jméno: admin

Heslo: admin

Po instalaci je také potřeba provést aplikování změn ve FreePBX. Kliknutím na oranžový pásek s textem *Apply Configuration Changes* a potvrzením dialogu *Continue with reload* se nastavení uloží do systému.

9.1.1 Aktualizace FreePBX

Nainstalovaná verze FreePBX je ve verzi 2.9.0.7. Jelikož je již vydána verze 2.10.0, provede se aktualizace na nejnovější stabilní verzi. V levém sloupci se přejde na *Module Admin* a poté se klikne *Check Online*. Objeví se tabulka vybízející k aktualizaci jednotlivých modulů. Celková aktualizace zajistí instalací modulu *2.10 Upgrade Tool*. Pro instalaci se vybere sekce *Action* a položka *Download and Install*. Stiskem tlačítka *Process* a potvrzením *Confirm* se provede instalace modulu.

V levém panelu přibyla nová sekce *2.10 Upgrade Tool*. Po přístupu na ni se klikne na tlačítko *UPGRADE NOW* a potvrdí se stiskem *OK*. Opět je nutné se vrátit do sekce *Module Admin*, kde se provede aktualizace. Stisknutím tlačítka *Check Online* a výběru položky *FreePBX Framework* a kliknutím na *Download and Upgrade to 2.10.1.9* se určí, co se má aktualizovat. Poté je postup stejný jako v případě instalace aktualizacího modulu. FreePBX začne stahovat z Internetu novou verzi a poté ji již nainstaluje do systému.

Aktualizace ještě není kompletní. Je nutné se zase vrátit do sekce *Module Admin* a zde znovu provést kontrolu jednotlivých modulů. Aktualizují se moduly *FreePBX ARI Framework*, *FreePBX Framework* a *Core*. Opětovným načtením stránky se klikne na *Apply Config*. Nová verze FreePBX s sebou přináší změnu grafického vzhledu.

Pro celkové dokončení instalace se musí zaktualizovat moduly v sekci *Admin* → *Module Admin*. Jedná se o *Custom Applications*, *Feature Code Admin*, *FreePBX FOP Framework*, *Recordings*, *Info Services*, *System Dashboard*, *Music on Hold* a *Voicemail*. Postup aktualizace je opět stejný. Provedené změny je nutné uložit stiskem červeného tlačítka *Apply Config*. Nyní je aktualizace kompletní.

9.2 Vytvoření IAX trunku k poskytovateli hlasových služeb

Softwarová pobočková ústředna je připojena k poskytovateli hlasových služeb pomocí trunku. Tento trunk zajišťuje přístup vybraných klapek do veřejné telefonní sítě. Tarifkaci má na starost poskytovatel, který zprostředkovává vyúčtování. Připojení do veřejné sítě se skládá z vytvoření trunku a nastavení příchozích a odchozích cest.

9.2.1 Vytvoření IAX trunku

K poskytovateli je vytvořen IAX trunk, který zaslal přihlašovací parametry (viz níže) pro připojení k jeho ústředně a samozřejmě také poskytl veřejná telefonní čísla pro volání.

Vytvoření trunku se provede přes položky *Connectivity* → *Trunks* a zvolí se *Add IAX2 Trunk*. V sekci *General Settings* se do pole *Trunk Name* napíše jméno trunku, které by mělo být výstižné (např. IAXtrunk_to_provider). Oddíl *Outgoing Settings* je nejdůležitější částí. Obsahuje údaje pro nastavení trunku. Zde je také opět pole *Trunk Name*, do které se napíše unikátní jméno trunku (např. IAX_trunk). Položka *PEER Details* obsahuje následující parametry:

```
username=uziv_iax
type=friend
secret=heslo
host=sip.poskytovatel.cz
requirecalltoken=no
qualify=yes
context=from-trunk
```

Položka *username* označuje uživatelské jméno a *secret* má přiřazeno heslo pro autentizaci na straně poskytovatele. Řádek *type* vyjadřuje vztah mezi školní ústřednou a vzdálenou ústřednou, který je typu *friend*. Adresa ústředny poskytovatele je zadána v položce *host*. U *requirecalltoken=no* není vyžadována Call Token Validation. Parametr *qualify=yes* zajišťuje kontrolu spojení mezi školní ústřednou a poskytovatelem. Pole *context* určuje, že se jedná o spojení z trunku.

Oddíl *Incoming Settings* slouží pro ověření příchozí komunikace a má pojmenovaný *USER Context* *uziv_iax*. V *USER Details* jsou vloženy parametry:

```
secret=heslo
type=user
context=from-trunk
```

Položka *secret* obsahuje heslo, které je stejné jako v nastavení výše. Parametr *type=user* vyjadřuje vztah a *context=from-trunk* je označení spojení.

Další polem, které je vyplněno v oddíle *Registration*, je položka *Register String* má registrační řetězec:

```
uziv_iax:heslo@sip.poskytovatel.cz
```

Tento registrační řetězec registruje trunk u poskytovatele. Stiskem tlačítka *Submit Changes* se uloží nastavení.

Příchozí směr vyžaduje Call Validation Token, který je na straně poskytovatele vypnut. Ústředna přesto vyžaduje vypnutí Tokenu, i když je již nastavení parametrů vloženo.

Proto se doinstaloval modul Asterisk IAX Setting a je umístěn v **Settings** → **Asterisk IAX Settings**. V položce **Other IAX Settings** se vloží parametr, který ignoruje tento Token ze všech adres:

```
calltokenoptional=0.0.0.0/0.0.0.0
```

Nastavení je uloženo přes tlačítko **Submit Changes**.

9.2.2 Nastavení odchozí cesty (Outbound Routes)

Odchozí cesta zajišťuje odeslání hovoru na vytvořený trunk k poskytovateli. Obsahuje vytvořené prefixy, které jsou povoleny pro danou odchozí cestu.

Route Name obsahuje pojmenování dané cesty. Srdcem celé routy jsou vzory předvoleb (**Dial Patterns**), které právě směřují vzory na trunk k poskytovateli. Jsou vytvořeny dvě odchozí cesty. Každá plní konkrétní úlohu. Je vytvořena cesta pro *Emergency* (tísňová volání), *Vnitrostatni* (volání v rámci státu).

Vzor předvoleb se skládá z **prepend**, **prefix**, **match pattern** a **CallerID**. V případě školní ústředny je využíváno pouze **match pattern**, jenž porovnává odchozí čísla.

```
(prepend) + prefix | [ match pattern / CallerID ]
```

Vzory předvoleb pro *Emergency* obsahují tíšňová volání, která jsou používána v České republice. Jsou to čísla 112, 150, 155, 156 a 158. U položky **Route Type** je zaškrtnuto **Emergency**, jenž označuje tíšňová volání.

```
( ) + | [112 / ]  
( ) + | [15[0568] / ]
```

Vnitrostatni obsahuje vzory pro všechna devítimístná národní čísla a pro vybraná čísla jako např. 14112 pro zjištění aktuálního času.

```
( ) + | [14112 / ]  
( ) + | [NXXXXXXXX / ]  
( ) + | [00. / ]
```

9.2.3 Nastavení příchozí cesty (Inbound Routes)

Nastavení příchozí cesty se nastaví přes **Connectivity** → **Inbound Routes**. Zobrazí se formulář pro přidání. Do pole **DID Number** se vloží přiřazené veřejné telefonní číslo. Nyní se musí toto číslo spárovat s danou klapkou. Toto se děje v položce **Set Destination**, kde se vybere **Extensions** a příslušná klapka. Kliknutím na **Submit** se uloží nastavení. Nyní je příchozí hovor z určeného veřejného čísla přeměřován na vybranou klapku.

9.3 Přidání klapky (Extensions)

Přidání nové klapky se ve FreePBX řeší v oddíle *Applications*. Jelikož se v této práci pracuje s protokolem SIP, vybere se typ zařízení *Generic SIP Device*. Zobrazí se formulář pro přidání nového SIPového zařízení.

Pro vytvoření nové klapky je potřeba vyplnit základní formulářová pole. Prvním polem je *User Extension*, do kterého se vloží číslo klapky (např. 101). Do *Display Name* se napíše jméno uživatele (např. Jan Novák), jenž je zobrazeno při volání. Položka *Outbound CID (CallerID)* obsahuje telefonní číslo, které se zobrazí při volání do veřejné sítě přes odchozí cestu a musí být vloženo, jinak hovor neproběhne. SIPové zařízení je chráněno heslem. Heslo se vkládá do pole *secret*, kde se napíše definované heslo. Obsahovat může alfanumerické znaky. V případě, že vytvořená klapka je umístěna za NATem, musí se vybrat u položky *nat* parametr *yes*. Pokud by se tak neudělalo, klapka by nebyla přístupná pro příchozí volání z Internetu. Ústředna již má v sobě implementovanou hlasovou schránku. Aktivace hlasové schránky se provedete výběrem volby *Enabled* u položky *Status*. Přístup do hlasové schránky je chráněn heslem. Heslo se zadává pouze jako čísla a vkládá se do pole *Voicemail Password*. Všechny provedené změny se uloží stiskem tlačítka *Submit*.

9.4 Instalace přídatných modulů ve FreePBX

FreePBX poskytuje uživateli základní funkce, které stačí pro základní chod softwarové pobočkové ústředny. Další služby či aplikace pro správu je možné přidat pomocí modulů. Nainstalovaný modul je přidán do navigační lišty, kde se pak provádí jeho konfigurace, případně poskytuje danou službu.

Instalace modulu je velmi podobná jako aktualizace FreePBX, jelikož i základní jádro je vlastní modul. Všechny moduly jsou umístěny v *Admin* → *Module Admin*. Kliknutím na příslušný modul se objeví nabídka, kde se v záložce *Action* zvolí *Download and Install*. Instalace se potvrdí stiskem *Process* a ještě jednou se potvrdí kliknutím na *Confirm*. Poté proběhne stažení a instalace. Dále se musí nainstalovaný modul zavést do systému. To se provede stiskem červeného tlačítka *Apply Config*.

Školní ústředna má instalovány tyto dodatečné moduly:

- **Asterisk Info** - Zobrazuje veškeré reporty týkající se ústředny. Jedná se o informace o SIP, IAX, hlasové schránce, registracích apod.
- **Asterisk Logfiles** - Tento modul zajišťuje výpis log soubor se jménem full, který je umístěn ve složce /var/log/asterisk. Ve výchozím stavu neprobíhá logování do souboru full. Musí se nejprve určit, co se bude zapisovat. Nastavení se vkládá do souboru /etc/asterisk/logger.conf, kam se vloží řádek full =>

`dtmf,notice,warning,error,debug`. Tento řádek říká, které události jsou zapisovány. Po zapsání konfigurace se zrestartuje logger příkazem `logger reload`. V případě, že se ještě neloguje do souboru, musí se zrestartovat i Asterisk příkazem `amportal restart`.

- **Print Extensions** - Modul má na starost přehledný výpis jednotlivých poboček, které jsou vytvořeny v ústředně. Jedná se o přehledný seznam telefonních čísel.
- **Asterisk IAX Settings** - Zpřístupňuje rozšířená nastavení pro protokol IAX. Je možné povolovat a měnit prioritu kodeků apod. Lze také nastavovat globální nastavení pro celý protokol.

9.5 Připojení studentské ústředny ke školní ústředně

Každý ze studentů si vytvoří na své pracovní stanici virtualizovanou softwarovou pobočkovou ústřednu. Studenská ústředna je připojena IAX trunkem ke školní ústředně, která poskytuje propojení mezi studenty. Trunk musí být vytvořen jak na straně studenta, tak i u výukové ústředny

9.5.1 Vytvoření IAX trunku na školní ústředně

Postup vložení nového IAX trunku se provede stejným způsobem jako u trunku k poskytovateli hlasových služeb. Rozdíl je pouze v zadaných parametrech. Opět se v **General Settings** zvolí **Trunk Name**. V následující sekci **Outgoing Settings** se zvolí jako **Trunk Name: student-peer**. Oddíl **PEER Details** obsahuje následující parametry:

```
username=skolni-user
type=peer
trunk=yes
secret=heslo
qualify=yes
host=studentska-pbx.fai.utb.cz
```

Popsány jsou pouze položky, které se liší od předchozího IAXového trunku. Parametr `type=peer` určuje, že se jedná o IAX2 trunk. Položka `trunk=yes` popisuje posílání kanálových dat v jednom trunku. Zajištění monitorování registrace spojení k cíli má na starost parametr `qualify=yes`.

Sekce **Incoming Settings** má v poli **USER Context** vloženo `student-user`. Oddíl **USER Details** má vloženy parametry:

```
type=user
secret=heslo
context=from-trunk
```

Položka `type=user` vyjadřuje, že se jedná o IAX2 spojení. Vložená nastavení trunku se kliknutím na **Submit Changes** uloží.

9.5.2 Nastavení odchozí cesty ze školní ústředny

Pojmenování odchozí cesty zajišťuje položka **Route Name**. Důležitou částí pro směrování hovorů z ústředny jsou {Dial Patterns}.

Pokud se chce uskutečnit hovor na studentskou ústřednu, která má klapky s čísly 2XX, musí být před konkrétní číslo vložena 0. Označuje, že se volá mimo ústřednu. Volané číslo je ve tvaru 02XX. Následující vzor řeší tento případ.

```
( ) + 0 | [2XX / ]
```

Odchozí hovor musí směřovat na příslušný trunk. V sekci **Trunk Sequence for Matched Routes** se vybere trunk na studentskou ústřednu. Nastavení cesty se uloží kliknutím na **Submit Changes**.

9.5.3 Vytvoření IAX trunku na studenské ústředně

Vytvoření trunku je stejné jako v předchozím případě. Změna je pouze v parametrech. Položka **Trunk Name** pojmenovává trunk. V oddíle **Outgoing Settings** je **Trunk Name** pojmenován **skolni-peer**. **PEER Details** obsahuje parametry:

```
username=student-user
type=peer
trunk=yes
secret=heslo
qualify=yes
host=sip.fai.utb.cz
```

Oddíl **Incoming Settings** má **USER Context** pojmenování **skolni-user**. Sekce **USER Details** obsahuje:

```
secret=heslo
type=user
context=from-trunk
```

Uložení nastavení se provede stiskem tlačítka **Submit Changes**.

9.5.4 Nastavení odchozí cesty ze studentské ústředny

Odchozí cesta má opět pojmenování v položce **Route Name**. Vzor je opět stejný jako v předchozím případě, jen je rozdíl v tom, že všechna odchozí čísla 1XX, jsou směřována do trunku. To znamená, že je vytočeno číslo 01XX, kde 0 značí odchozí hovor z ústředny.

```
( ) + 0 | [1XX / ]
```

V sekci **Trunk Sequence for Matched Routes** se vybere příslušný odchozí trunk na školní ústřednu. Uložení se provede stiskem **Submit Changes**.

10 ZABEZPEČENÍ ÚSTŘEDNY

Ústředna je zabezpečena v několika oblastech. Základním zabezpečením je ochrana operačního systému před neoprávněným vniknutím. Tuto oblast má na starost vytvořený firewallový skript, který povoluje spojení pouze pro dané IP adresy a porty. Spojení z jiných IP adres se neuskuteční. Popis funkce skriptu je popsán v předchozích kapitolách.

Administrace celého serveru probíhá vzdáleně přes SSH. Uživatel se přihlašuje na server pomocí klíče a tímto se ověří jeho identita. Všechna komunikace mezi serverem a uživatelem je šifrována.

10.1 Změna výchozích hesel

Instalace Asterisku a FreePBX se provádí s výchozími hesly. Jedním ze základních bezpečnostních prvků je změna a volba dostatečně silného hesla. Hesla se mění v konfiguračních souborech pro uživatele *asteriskuser*, který je používán pro MySQL. Změna se provede pod uživatelem *root* a zadáním následujících příkazů:

```
# NEWPASSWORD='heslo';
# sed -i "s/\(^AMPDBPASS=*\)\"(.*)\"/\1$NEWPASSWORD/"
  /etc/amportal.conf
# sed -i "s/\(^password = *\)\"(.*)\"/\1$NEWPASSWORD/"
  /etc/asterisk/cdr_mysql.conf
# sed -i "s/\(^AMPDBPASS = *\)\"(.*)\"/\1$NEWPASSWORD/"
  /etc/asterisk/extensions_additional.conf
# sed -i "s/\($amp_conf\[ 'AMPDBPASS '\]\t= '*\)\"(.*)\"/\
  1$NEWPASSWORD';
  /" /etc/freepbx.conf
# echo "SET PASSWORD FOR 'asteriskuser'@'localhost' =
  PASSWORD('$NEWPASSWORD');" | mysql -u root -p
```

[27]

Další krokem je změna výchozích hesel u Asterisk Manageru. Příkazy pro provedení změny jsou:

```
# NEWPASSWORD='heslo';
# sed -i "s/\(^secret = *\)\(.*\)/\1$NEWPASSWORD/"
    /etc/asterisk/manager.conf
# sed -i "s/\(^AMPMGRPASS = *\)\(.*\)/\1$NEWPASSWORD/"
    /etc/asterisk/extensions_additional.conf
# sed -i "s/\(^AMPMGRPASS=*)\(.*\)/\1$NEWPASSWORD/"
    /etc/amportal.conf
# asterisk -r -x "module reload manager"
# echo "UPDATE freepbx_settings SET value='$NEWPASSWORD'
    WHERE keyword='AMPMGRPASS';" | mysql -u root -p asterisk
```

[27]

Nyní jsou změněna všechna hesla, která se používala při instalaci. Na řadě je tedy změna hesel, která se týká samotného FreePBX. Změna hesla administrátora se mění v **Admin** → **Administrators**, kde se v pravé části vybere položka **admin**.

Ostatní hesla související s provozem FreePBX se mění v sekci **Settings** → **Advanced Settings**. Změna hesel je nutná u **FOP Password** a **User Portal Admin Password**.

11 ŘEŠENÍ LABORATORNÍCH ÚLOH

Vytvořené laboratorní úlohy slouží studentům k seznámení se s principem VoIP komunikace. Student zjistí jak taková ústředna funguje, uvede ji do provozu a také jakým způsobem ji zabezpečit proti případným útokům.

11.1 Instalace softwarové pobočkové ústředny Asterisk

Laboratorní úloha č.1 se věnuje instalaci operačního systému GNU/Linux Debian, softwarové pobočkové ústředny Asterisk a grafického webového rozhraní FreePBX.

První část popisuje instalaci a nastavení Debianu. Po instalační nastavení se týkají aktualizace systému, přiřazení uživatele do skupiny sudo (při vzdáleném přihlášení není povolen uživatel **root**), nastavení SSH pro vzdálený přístup na server a nastavení pevné IP adresy pro připojení do sítě LAN.

11.2 Zabezpečení ústředny

Laboratorní úloha č.2 je zaměřena na zabezpečení samotného operačního systému pomocí firewallového skriptu, který je základním bezpečnostním prvek. Student se seznámí, jak takový skript vypadá a k čemu slouží. Dalé je popsán postup pro změnu výchozích hesel v konfiguračních souborech a v webovém rozhraní FreePBX.

11.3 Nastavení ústředny

Poslední laboratorní úloha popisuje nastavení softwarové pobočkové ústředny. Část je věnována vytvoření a nastavení trunku k poskytovateli hlasových služeb, další část je zaměřena na vytvoření trunku mezi studentskou a školní ústřednou. Popsáno je také i nastavení příchozích a odchozích cest, které jsou důležité pro směrování hovorů na ústřednu a mimo ni.

ZÁVĚR

Tato práce se zabývala vytvořením softwarové pobočkové ústředny. Byla popsána instalace, nastavení a zabezpečení operačního systému GNU/Linux Debian, softwaru pobočkové ústředny Asterisk a webového rozhraní FreePBX. Úkolem bylo vytvořit školní ústřednu, která bude sloužit studentům v rámci výuky předmětu Telekomunikační systémy. Součástí bylo také sestavení laboratorních úloh, které se věnovaly jednotlivým fázím provozu ústředny (instalace, nastavení a zabezpečení). Výuková PBX byla připojena k poskytovateli hlasových služeb.

Instalace operačního systému Debian probíhala standardním způsobem a v průběhu se nevyskytl žádný závažný problém. Přímo z instalátoru bylo pouze nutné nainstalovat SSH server. Ostatní parametry instalace zůstaly nastaveny na výchozí hodnotě. Po instalaci bylo provedeno nastavení operačního systému (aktualizace, přiřazení uživatele do skupiny *sudo*, nastavení SSH, přiřazení veřejné IP adresy) a zprovoznění firewallového skriptu, který povoloval připojení k serveru pouze daným IP adresám a portům.

Dalším krokem byla instalace softwaru pobočkové ústředny Asterisk. Instalace probíhala ze zdrojových kódů, které se musely předem kompilovat. Při kompilaci byly nastaveny parametry, které ovlivnily výchozí nastavení. U testovací instalace ve Virtualboxu se vyskytl problém, kdy Asterisk nešel spustit a upozorňoval na chybu. Příčinou byla kompilace softwaru na nejnovějším hardwaru a kompilace byla nastavena tak, aby se přizpůsobila danému hardwaru pro lepší optimalizaci procesoru. Po odškrtnutí příslušné položky se Asterisk spustil korektně.

Poslední částí byla instalace webového rozhraní FreePBX. Před samotným procesem bylo nutné provést instalaci důležitých balíčků, které byly nezbytné pro správný chod FreePBX. Byla provedena také nezbytná nastavení a poté byla spuštěna instalace. Závěrečným krokem byla aktualizace FreePBX na nejnovější verzi.

Po instalaci celé ústředny byla provedena změna veškerých výchozích hesel, jak u Asterisku tak i u webového rozhraní FreePBX. Protože ústředna je připojena k poskytovateli hlasových služeb, je v této práci popsáno její připojení do veřejné telefonní sítě. Nastavení příchozích a odchozích hovorů bylo vytvořeno s pomocí rozhraní FreePBX, kde se určené klapce přiřadilo veřejné telefonní číslo. Bylo provedeno také otestování připojení studentské ústředny ke školní ústředně, v rámci kterého byly provedeny zkušební hovory. Nastavení připojení je velmi podobné jako připojení do veřejné telefonní sítě.

V průběhu vytváření IAX trunku mezi poskytovatelem a školní ústřednou došlo k situaci, že žádné příchozí hovory z veřejné telefonní sítě nebyly uskutečněny. Naproti tomu všechny odchozí hovory probíhaly bez problémů. Z počátku byl problém s registrací, protože na ústřednu nebyl poslán žádný hovor a příchozí komunikace neprobíhala. Po opravě tohoto problému se již příchozí hovor dostal na ústřednu, ale byl odmítnut,

protože se dožadoval zapnutí ověření pomocí Call Token Validation na straně školní ústředny, i když v parametrech trunku je vypnutý. Na straně poskytovatele bylo toto ověření vypnuté a proto musela být nastavena výjimka v globálním nastavení IAX trunku. Poté byl hovor opět odmítnut, a to z důvodu, že se nemohl ověřit příchozí směr. Proto byly přidány příslušné parametry do sekce *Incoming Settings*. Volání z veřejné telefonní sítě bylo již v pořádku a hovor se uskutečnil.

Nasazení softwarové pobočkové ústředny Asterisk v kombinaci s webovým rozhraním FreePBX bylo správným krokem. V průběhu provozu byla ústředna stabilní a nebyl zaznamenán žádný hardwarový či softwarový problém. Oba projekty jsou výborně zdokumentované. Zvolená varianta operačního systému GNU/Linux Debian, Asterisku a FreePBX je mimořádně vhodná pro výuku v oblasti VoIP komunikace.

ZÁVĚR V ANGLIČTINĚ

The subject of this work was to create a software PBX. This work contains a description of the installation, setup and security of the GNU/Linux Debian operating system and FreePBX web interface. The task was to create a school exchange, which will serve for students in the lessons of telecommunications systems. The part of the work was to create laboratory exercise focusing on particular stage of exchange operation (installation, configuration and security). The testing PBX was connected to voice services provider.

The installation of Debian OS proceeded in a standard way and without serious problems. The SSH server was installed directly from the installer. Other installation parameters remained in default settings. After installation the operating system was configured (update, assigning a user to *sudo* group, SSH setting, assigning a public IP address) and a firewall script was enabled. This script allowed access to the server only selected IP addresses and ports.

The next step was an installation of the PBX Asterisk software. The installation ran from source codes that had to be compiled. The key parameters, affecting the default settings, were set during the compilation of the source codes. The problem occurred during test installation in the Virtualbox - Asterisk did not start and pointed out the error. The reason was a compilation of the software on the latest hardware when the compilation was adjusted to accommodate that hardware for better processor optimization. After unchecking the item Asterisk started correctly.

The last part was the installation of the FreePBX web interface. The installation of important packages, that were necessary for FreePBX running, had to be performed before the installation process itself. Necessary settings were made, and then the installation started. The final step was to update to the latest version of FreePBX.

After the exchange installation, all default passwords were changed in Asterisk software as well as in FreePBX web interface. The exchange is connected to the voice services provider and therefore the work has described this connection to the PSTN. The setting of incoming and outgoing calls was created using the FreePBX interface, through which a public telephone number is assigned to an appropriate extension. There were also established a testing connection between the testing exchange and school exchange and test calls were executed. Connection setting is very similar to a connection to the PSTN.

During the creation of an IAX trunk between a provider and school exchange the following situation occurred: no incoming calls from the PSTN were made. In contrast, all outgoing calls were carried out without any problems. At the beginning, there was a problem with a registration, because the panel didn't send any incoming call and the

communication did not work. After the correction of this problem, an incoming call passed to the PBX, but it was rejected because it demanded turning on the verification using the Call Token Validation on the school exchange, even though the appropriate parameters in the trunk were turned off. This verification was disabled on the provider's side and therefore an exception had to be set in the global setting of the IAX trunk. After that the call was rejected again, because of it could not verify the incoming direction. Therefore, the appropriate parameters were added to the *Incoming Settings* section. After that the calls from the PSTN have been successful.

The deployment of the Asterisk software in combination with FreePBX web interface was the right decision. During operation, the exchange was stable and no hardware or software problem was occurred. Both projects are well documented. The selected variant of the GNU/Linux Debian operating system, Asterisk and FreePBX is very suitable for teaching in the field of VoIP communications.

SEZNAM POUŽITÉ LITERATURY

- [1] ŠIMÁK, Boris. Principy zpracování hlasu v klasické a IP telefonii. In: *Teorie a praxe IP telefonie* [online]. 2004 [cit. 2013-03-25]. Dostupné z: <http://www.phonet.cz/archiv/dok_osta/ipt-2004_Principy_zprac_hlasu.pdf>
- [2] JIROUŠEK, Radim, Jiří IVÁNEK, Petr MÁŠA a Norbert VANĚK. *Principy digitální komunikace*. Vyd. 1. Voznice: Leda, 2006, 309 s. ISBN 80-733-5084-X.
- [3] DAVÍDEK, Vratislav, Miloš LAIPERT a Miroslav VLČEK. *Analogové a číslicové filtry*. Vyd. 2. Praha: Vydavatelství ČVUT, 2004, 345 s. ISBN 80-010-3026-1.
- [4] HLAVÁČ, Václav a Miloš SEDLÁČEK. *Zpracování signálů a obrazů*. Vyd. 2. Praha: ČVUT, 2005, 255 s. ISBN 80-010-3110-1.
- [5] Diskrétní signál. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2013 [cit. 2013-03-26]. Dostupné z: <http://cs.wikipedia.org/wiki/Diskr%C3%A9tn%C3%AD_sign%C3%A1l>
- [6] VLČEK, Karel. *Přednáška č.1 z předmětu Digitální komunikace*. Zlín, 2012.
- [7] ČERNOCKÝ, Honza. *Zpracování řečových signálů* [online]. Brno, 6. prosince 2006 [cit. 2013-03-28]. Dostupné z: <http://www.fit.vutbr.cz/study/courses/ZRE/public/opora/zre_opora.pdf>. Studijní opora. Fakulta informačních technologií, VUT v Brně.
- [8] VODRÁŽKA, Jiří a Ivan PRAVDA. *Principy telekomunikačních systémů*. Vyd. 1. Praha: ČVUT, 2006, 130 s. ISBN 80-010-3366-X.
- [9] VODRÁŽKA, Jiří a Martin HAVLAN. *Přenosové systémy: sítě a zařízení SDH a jejich návrh*. 3., přeprac. vyd. V Praze: České vysoké učení technické, 2009, 205 s. ISBN 978-80-01-04217-5.
- [10] MEGGELEN, Jim Van, Leif MADSEN a Jared SMITH. *Asterisk: the future of telephony*. 2nd ed. Beijing: O'Reilly, 2007, 574 s. ISBN 05-965-1048-9.
- [11] HOLUB, Jan. Měření a hodnocení QoS v IP telefonii. In: *Teorie a praxe IP telefonie* [online]. 2004 [cit. 2013-04-03]. Dostupné z: <http://www.ip-telefon.cz/archiv/dok_osta/ipt-2004_Mereni_hodnoceni_QoS.pdf>
- [12] VOZŇÁK, Miroslav. *Voice over IP*. 1. vyd. Ostrava: VŠB - Technická univerzita Ostrava, 2008, 176 s. ISBN 978-80-248-1828-3.
- [13] VOZŇÁK, Miroslav. Technické principy IP telefonie. In: *Teorie a praxe IP telefonie* [online]. 2004 [cit. 2013-04-04]. Dostupné z: <http://www.ip-telefon.cz/archiv/dok_osta/ipt-2004_Principy_IPtel.pdf>

- [14] CAMP, Ken. IP telephony demystified. New York: McGraw-Hill, c2003, xv, 254 p. McGraw-Hill networking professional. ISBN 00-714-0670-0.
- [15] SIP [IpTelWiki]. *CESNET* [online]. 2012, 2012/03/11 22:14 [cit. 2012-11-13]. Dostupné z: <<https://sip.cesnet.cz/cs/protokoly/sip>>
- [16] VOZŇÁK, Miroslav. Telefonní ústředny Asterisk. In: *Teorie a praxe IP telefonie* [online]. 2008 [cit. 2013-04-12]. Dostupné z: <http://www.ip-telefon.cz/archiv/dok_osta/ipt-2008_Telefonni_ustredny_Asterisk.pdf>
- [17] WIJA, Tomáš, David ZUKAL a Miroslav VOZŇÁK. *Asterisk a jeho použití* [on-line]. 2005 [cit. 2013-04-13]. Dostupné z: <<http://archiv.cesnet.cz/doc/techzpravy/2005/voip/asterisk.pdf>>
- [18] MADSEN, Leif. Contexts, Extensions, and Priorities. In: *Asterisk Project Wiki* [online]. 2011 [cit. 2013-04-15]. Dostupné z: <<https://wiki.asterisk.org/wiki/display/AST/Contexts%2C+Extensions%2Cand+Priorities>>
- [19] FreePBX - voip-info.org. *Voip-info.org* [online]. 2012, Fri 23 of Mar, 2012 (18:31) [cit. 2013-04-15]. Dostupné z: <<http://www.voip-info.org/wiki/view/freePBX>>
- [20] Welcome | FreePBX. *FreePBX* [online]. 2013 [cit. 2013-04-15]. Dostupné z: <<http://www.freepbx.org/>>
- [21] Installation | FreePBX. *FreePBX* [online]. 2013 [cit. 2013-04-15]. Dostupné z: <<http://www.freepbx.org/support/freepbx-terms/documentation/installation-0>>
- [22] VANĚK, Tomáš. Bezpečnost v oblasti VoIP. In: *Teorie a praxe IP telefonie* [online]. 2008 [cit. 2013-04-15]. Dostupné z: <http://www.ip-telefon.cz/archiv/dok_osta/ipt-2008_Bezpecnost_VoIP.pdf>
- [23] VANĚK, Tomáš. Bezpečnost provozu VoIP. In: *Teorie a praxe IP telefonie* [online]. 2006 [cit. 2013-04-16]. Dostupné z: <www.ip-telefon.cz/archiv/dok_osta/ipt-2006_Bezpecnost_VoIP.pdf>
- [24] PETER, Tomáš a Dalibor MICHALEC. IPsec. *VŠB / Katedra informatiky FEI VŠB-TUO* [online]. 2013 [cit. 2013-04-16]. Dostupné z: <<http://www.cs.vsb.cz/grygarek/TPS-0304/projekty0304/ipsec/ipsec.html>>
- [25] ZRTP - voip-info.org. In: *Voip-info.org* [online]. 2013 [cit. 2013-04-16]. Dostupné z: <<http://www.voip-info.org/wiki/view/ZRTP>>

-
- [26] SRTP - voip-info.org. *Voip-info.org* [online]. 2013 [cit. 2013-04-16]. Dostupné z: Dostupné z: <<http://www.voip-info.org/wiki/view/SRTP>>
- [27] Debian | Asterisk + FreePBX Complete Installation Guide. KEER, Kulvinder. *Infinity* [online]. 2012, March 5, 2012 [cit. 2013-05-01]. Dostupné z: <<http://blog.keer.info/debian-asterisk-freepbx-complete-installation-guide/>>
- [28] Umíte vyjmenovat všechny známé odpovědi SIP?. *IP PBX: 3CX IP PBX pro Windows PBX Pobočková telefonní systém* [online]. 2012 [cit. 2013-05-05]. Dostupné z: <<http://www.3cx.cz/voip-sip/sip-responses.html>>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

3DES	Triple Data Encryption Standard
A/D převodník	Analogově číslicový převodník
ACD	Automatic Call Distribution
ACELP	Algebraic - code - excitation
ADPCM	Adaptive Differential Pulse Code Modulation
AES	Advanced Encryption Standard
AH	Authentication Header
API	Application Programming Interface
BRI	Basic Rate Interface
BSD	Berkeley Software Distribution
CELP	Code Excited Linear Prediction
CID	CallerID
CPU	Central Processing Unit
cRTP	compressed RTP
D/A převodník	číslicově analogový převodník
DAM	Diagnostic Acceptability Measure
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DPCM	Differential Pulse Code Modulation
DRT	Diagnostic Rhyme Test
DSP	Digital Signal Processing
DTMF	Dual - tone multi - frequency
EAPoL	EAP over LAN
ESP	Encapsulating Security Payload
GNU GPL	General Public Licence
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
IAX	Inter - Asterisk eXchange
IEEE	The Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
iLBC	Internet Low Bitrate Code
IP	Internet Protocol
IPsec	IP security
ISDN	Integrated Services Digital Network
ITU	International Telecommunication Union
IVR	Interactive Voice Responce

LAN	Local Area Network
LP	Linear Prediction
LTP	Long Term Prediction
LVM2	Logical Volume Management
MD - 5	Message-Digest - 5
MGCP	Media Gateway Control Protocol
MMUSIC	Multiparty Multimedia Session Control
MOS	Mean Opinion Score
MP - MLQ	Multi - pulse Maximum Likelihood Quantization
MPE	Multi - Pulse Excited
MSC	Multi - path Search Coding
NAT	Network Address Translation
NGN	Next Generation Network
PBX	Private Branch Exchange
PCM	Pulse Code Modulation
PGP	Pretty Good Privacy
PKI	Public Key Infrastructure
POTS	Plain Old Telephone Service
PRI	Primary Rate Interface
PSK	Pre - shared key
PSQM	Perceptual Speech Quality Measure
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RAID	Redundant Array of Inexpensive/Independent Disks
REL P	Residually Excited Linear Prediction
RPE	Regular Pulse Excitation
RTCP XR	Real - Time Control Protocol Extended Reports
RTP	Real - time Transport Protocol
RTPC	Real - Time Control Protocol
S/MIME	Secure/Multipurpose Internet Mail Extensions
SAS	Short Authentication String
SDP	Session Description Protocol
SHA - 1	Secure Hash Algorithm
SIP	Session Initiation Protocol
SIPS	SIP Secure
SMTP	Simple Mail Transfer Protocol
SNR	Signal-to-noise ratio

SRTP	Secure RTP
SSH	Secure Shell
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TDM	Time Division Multiplex
TFTP	Trivial File Transfer Protocol
TLS	Transport Layer Security
UA	User Agents
UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
VAD	Voice Activity Detector)
VLAN	Virtual LAN
VoFR	Voice over Frame Relay
VoIP	Voice over IP
VVLAN	Voice VLAN

SEZNAM OBRÁZKŮ

Obr. 1. Řečové ústrojí [1, s. 41]	12
Obr. 2. Vzorkovaný signál [5]	14
Obr. 3. Kvantovaný signál [5]	15
Obr. 4. Kompresní charakteristika pro PCM telefon [6, s. 16]	15
Obr. 5. Model SIPu [14, s. 82].....	27
Obr. 6. Architektura Asterisku [16, s. 28]	36
Obr. 7. Ukázka aplikace FreePBX	39
Obr. 8. Výběr instalace SSH serveru z instalátoru	47
Obr. 9. Nastavení jednotlivých položek před kompilací.....	53
Obr. 10.Instalace SSH serveru z instalátoru.....	87
Obr. 11.Výběr jednotlivých položek před kompilací	91

SEZNAM TABULEK

Tab. 1. Přehled parametrů kodeků [1, s. 46].....	19
Tab. 2. Hodnoty pro MOS [7, s. 74].....	21
Tab. 3. Hardwarové nároky Asterisku [10, s. 12].....	32

SEZNAM PŘÍLOH

- P I. SIP kódy odpovědí
- P II. Firewallový skript zabezpečující server
- P III. Skript pro spuštění Asterisku po startu operačního systému
- P IV. Laboratorní úloha č. 1 - Instalace softwarové pobočkové ústředny Asterisk
- P V. Laboratorní úloha č. 2 - Zabezpečení ústředny
- P VI. Laboratorní úloha č. 3 - Nastavení ústředny

PŘÍLOHA P I. SIP KÓDY ODPOVĚDÍ

[28]

1xx = informační odpovědi

- 100 Trying (Pokus)
- 180 Ringing (Vyzvánění)
- 181 Call Is Being Forwarded (Hovor je přesměrován)
- 182 Queued (Fronta)
- 183 Session Progress (Postup relace)

2xx = odpovědi o dosažení úspěchu

- 200 OK
- 202 accepted: Used for referrals (přijat: Použití pro doporučení)

3xx = odpovědi o přesměrování

- 300 Multiple Choices (Více voleb)
- 301 Moved Permanently (Trvale přemístěn)
- 302 Moved Temporarily (Dočasně přemístěn)
- 305 Use Proxy (Použití proxy)
- 380 Alternative Service (Alternativní služba)

4xx = selhání příkazu

- 400 Bad Request (Nesprávný příkaz)
- 401 Unauthorized: Used only by registrars. Proxys should use proxy authorization 407 (Neautorizovaný: Používají pouze registrátoři. Při využití proxy by se měla použít autorizace proxy 407).
- 402 Payment Required (Reserved for future use) (Požadována platba (rezervováno pro budoucí použití))
- 403 Forbidden (Zakázáno)
- 404 Not Found: User not found (Není nalezen: Uživatel nebyl nalezen)

- 405 Method Not Allowed (Metoda není povolena)
- 406 Not Acceptable (Není přijatelné)
- 407 Proxy Authentication Required (Požadovaná autentizace proxy)
- 408 Request Timeout: Couldn't find the user in time (Časový limit příkazu: Uživatel se nepodařilo najít včas)
- 410 Gone: The user existed once, but is not available here any more. (Je pryč: Uživatel kdysi existoval, ale již není k dispozici.)
- 413 Request Entity Too Large (Entita příkazu je příliš velká)
- 414 Request-URI Too Long (URI příkazu je příliš dlouhý)
- 415 Unsupported Media Type (Typ média není podporován)
- 416 Unsupported URI Scheme (Plán URI není podporován)
- 420 Bad Extension: Bad SIP Protocol Extension used, not understood by the server (Nesprávná přípona: Je použita nesprávná přípona protokolu SIP, pro server není srozumitelná)
- 421 Extension Required (Je požadována přípona)
- 423 Interval Too Brief (Interval je příliš krátký)
- 480 Temporarily Unavailable (Dočasně není k dispozici)
- 481 Call/Transaction Does Not Exist (Hovor/transakce neexistuje)
- 482 Loop Detected (Detekována smyčka)
- 483 Too Many Hops (Příliš mnoho skoků)
- 484 Address Incomplete (Adresa není úplná)
- 485 Ambiguous (Nejasný)
- 486 Busy Here (Zaneprázdněný)
- 487 Request Terminated (Příkaz ukončen)
- 488 Not Acceptable Here (Nepřijatelný)
- 491 Request Pending (Příkaz čeká)
- 493 Undecipherable: Could not decrypt S/MIME body part (Nerozluštitelný: Nepodařilo se rozluštit část těla S/MIME)

5xx = chyby serveru

- 500 Server Internal Error (Vnitřní chyba serveru)
- 501 Not Implemented: The SIP request method is not implemented here (Neimplementováno: Příkaz/metoda SIP zde není implementována)
- 502 Bad Gateway (Nesprávná brána)
- 503 Service Unavailable (Služba není k dispozici)
- 504 Server Time-out (Časová limit serveru)
- 505 Version Not Supported: The server does not support this version of the SIP protocol (Verze není podporována: Server nepodporuje tuto verzi protokolu SIP)
- 513 Message Too Large (Zpráva je příliš velká)

6xx = globální selhání

- 600 Busy Everywhere (Vše je zaneprázdněno)
- 603 Decline (Odmítnutí)
- 604 Does Not Exist Anywhere (Nikde neexistuje)
- 606 Not Acceptable (Nepřijatelný)

PŘÍLOHA P II. FIREWALLOVÝ SKRIPT ZABEZPEČUJÍCÍ SERVER

```
#!/bin/sh

#####
# Firewall povoluje komunikaci jen s pocitaci, ktere maji povolenou IP.
# Komunikace je povolena pouze na povolene sluzby poskytovane serverem.
#####

# hlavicka potrebna pro spusteni pomoci /etc/init.d ve Squeeze
### BEGIN INIT INFO
# Provides:          firewall
# Required-Start:    $local_fs $network
# Required-Stop:     $local_fs $network
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: firewall
### END INIT INFO

# porty Asterisk UDP - hlas
HLAS="10000:20000"
IAX2="4569"

# definovani povolenych IP a sluzeb
POVOLENE_IP="10.0.0.0/8"
POVOLENE_IP_APT="10.0.0.0/8"
POVOLENE_SLUZBY_TCP="ssh sip http rtsp ftp ftp-data"
POVOLENE_SLUZBY_UDP="rtsp sip $HLAS $IAX2"

# spusteni a vlozeni pravidel
start_firewall () {

    # definovani zakladnich pravidel (policy) pro tabulku filter
    iptables -P INPUT DROP
    iptables -P OUTPUT ACCEPT

    # povoleni pravidla pro praci s DNS - nslookup
    iptables -A INPUT -p udp --sport 53 -s 195.178.88.66 -j ACCEPT

    # povoleni pravidla pro aktualizace - apt a freepbx (mirror)
```

```

for IP in $POVOLENE_IP_APT
do
    iptables -A INPUT -p tcp --sport 80 -s $IP -j ACCEPT
done

# definovani pravidel
# povoleni pro spojeni TCP
for SLUZBY in $POVOLENE_SLUZBY_TCP
do
    for IP in $POVOLENE_IP
    do
        iptables -A INPUT -p tcp --dport $SLUZBY -s $IP -j ACCEPT
    done
done

# povoleni pro spojeni UDP
for SLUZBY in $POVOLENE_SLUZBY_UDP
do
    for IP in $POVOLENE_IP
    do
        iptables -A INPUT -p udp --dport $SLUZBY -s $IP -j ACCEPT
    done
done

# povoleni pristupu z localhosta vsechno
iptables -A INPUT -j ACCEPT -i lo

# povoleni ICMP ze vseh stroju
iptables -A INPUT -p icmp -j ACCEPT

# co neni povoleno na TCP, dostane RST, na UDP ICMP Port Unreachable
iptables -A INPUT -i eth0 -p tcp -j REJECT --reject-with tcp-reset
iptables -A INPUT -i eth0 -p udp -j REJECT --reject-with icmp-port-unreachable
}

# pri zastaveni smaze puvodni pravidla a vlozi zakladni pravidla, vse povoleno
stop_firewall () {
    iptables -t filter -F

```

```
iptables -t filter -P INPUT ACCEPT
iptables -t filter -P OUTPUT ACCEPT
}

# rozhrani pri spousteni pres BASH
case "$1" in
  start)
    echo "Spoustim firewall a fail2ban!"
    start_firewall
    /etc/init.d/fail2ban start
    ;;

  stop)
    echo "Zastavuji firewall a fail2ban!!!"
    stop_firewall
    /etc/init.d/fail2ban stop
    ;;

  restart)
    echo "Restartuji firewall a fail2ban!!"
    stop_firewall
    start_firewall
    /etc/init.d/fail2ban restart
    ;;

  *)
    echo "Prikaz pro spusteni: firewall.sh start"
    echo "Prikaz pro zastaveni: firewall.sh stop"
    echo "Prikaz pro restart: firewall.sh restart"
    ;;

esac
```

PŘÍLOHA P III. SKRIPT PRO SPUŠTĚNÍ ASTERISKU PO STARTU OPERAČNÍHO SYSTÉMU

```
#!/bin/sh
# /etc/init.d/amportal-startup
#

### BEGIN INIT INFO
# Provides:          Asterisk
# Required-Start:    $remote_fs $syslog $all
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Start Asterisk at boot time
# Description:       Enable Asterisk.
### END INIT INFO

PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

export PATH

case "$1" in
  start)
    amportal start
    ;;
  stop)
    amportal stop
    ;;
  *)
    echo "Usage: /etc/init.d/amportal-startup {start|stop}"
    exit 1
    ;;
esac

exit 0
```

PŘÍLOHA P IV. LABORATORNÍ ÚLOHA Č. 1 - INSTALACE SOFTWARE PŘI PŘÍLOZE ÚSTŘEDNÍ ASTERISK

Úkoly

1. Seznamte se zadáním a instalačním postupem u laboratorní úlohy č.1.
2. Nainstalujte operační systém GNU/Linux Debian a nastavte jej.
3. Proveďte instalaci PBX Asterisk.
4. Proveďte instalaci webové grafické nástavby FreePBX.

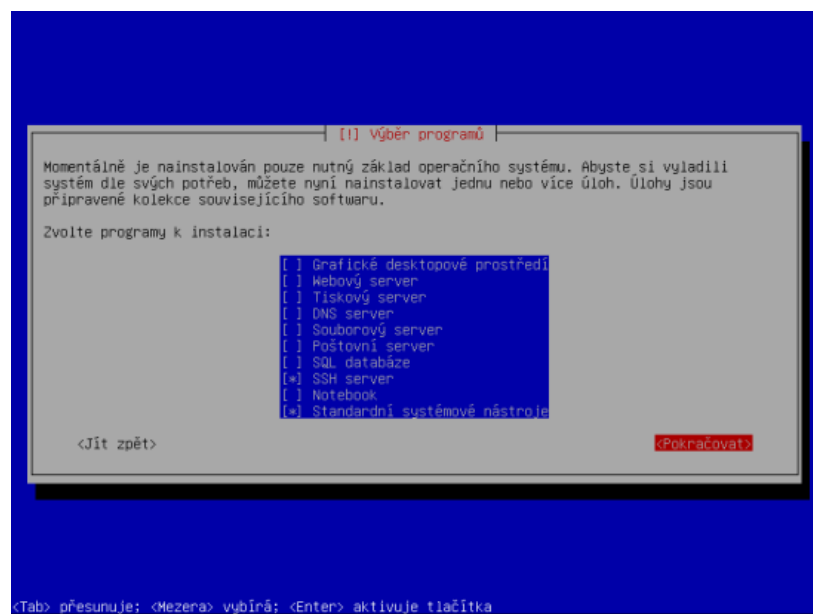
INSTALACE A NASTAVENÍ OPERAČNÍHO SYSTÉMU GNU/LINUX DEBIAN

Instalace operačního systému

Pro instalaci je nutné mít přichystán obraz GNU/Linux Debian Squeeze (např. debian-6.0.6-amd64-i386-netinst.iso). Celá instalace bude probíhat ve virtuální prostředí za pomoci virtualizačního nástroje Virtualbox nebo VMware player.

Virtuální stroj musí být připojen do školní sítě LAN a adresu získává z DHCP serveru. Velikost diskového prostoru je minimálně 8GB a 1GB paměti RAM. Instalační obraz se nastaví jako bootovací a ihned po startu se načte instalátor Debianu.

Instalace probíhá podle pokynů instalátoru. Je nutné vytvořit dva uživatele (root – práva superuživatele, uziv-tks – práva běžného uživatele). Veškeré rozdělení disku je ponecháno na systému. Pouze v případě instalace nabídky výběru programů je z instalátoru nainstalován SSH server pro vzdálený přístup (Obr. 1)



Obr. 10. Instalace SSH serveru z instalátoru

Po instalační nastavení operačního systému

Aktualizace systému

Pro základní důležitá nastavení je využíván uživatel *root*. Veškerá nastavení jsou nutná pro správný a bezpečný chod operačního systému. Nejdříve se provede aktualizace systému pomocí příkazu:

```
# apt-get update && apt-get upgrade
```

Přiřazení uživatele do skupiny sudo

Dalším krokem je přiřazení uživatele *uziv-tks* do skupiny *root*, která má práva pro správu celého systému. Tento uživatel přistupuje k systémovým úkonům a nastavením přes program **sudo**, který pro potvrzení akce vyžaduje heslo uživatele. Výchozí instalace Debianu neobsahuje balíček **sudo** a je nutné jej ručně nainstalovat příkazem:

```
# apt-get install sudo
```

Přiřazení uživatele se provede příkazem:

```
# visudo
```

Otevře se soubor */etc/sudoers* obsahující nastavení. Do oddílu **User privilege specification** se vloží následující řádky, které zajistí přístup přes **sudo**:

```
uziv-tks    ALL=(ALL:ALL) ALL
```

Nastavení SSH

Jelikož se k serveru bude přistupovat vzdáleně, je nutné nastavit přístup přes SSH. Konfigurace SSH serveru se nachází v */etc/ssh/sshd_config*. V souboru *usshd_config* je nutno upravit následující

```
PermitRootLogin no  
PasswordAuthentication yes  
X11Forwarding no
```

Položka **PermitRootLogin no** zakazuje přihlásit se jako *root* přes SSH. Příkaz, který zajišťuje ověření hesla při přihlášení je **PasswordAuthentication yes**. Poslední je **X11Forwarding no**, kdy se zakáže tunelování X11.

Přiřazení pevné IP adresy

V některých případech (připojení k poskytovateli hlasových služeb – veřejná IP adresa) je nutné mít nastavenou pevnou IP adresu. Nastavení síťového rozhraní se nachází v souboru */etc/network/interfaces*. Uvnitř souboru se provedou změny. Ve výchozím stavu je nastaveno získání IP adresy z DHCP. Tyto následující položky se ze souboru odstraní nebo se zakomentují, aby nebyly aktivní.

```
auto eth0
  allow-hotplug eth0
  iface eth0 inet dhcp
```

Místo nich se vloží následující řádky, které přiřadí pevnou IP adresu síťovému rozhraní. Zápis je následující:

```
auto eth0
  iface eth0 inet static
  address 192.168.1.2
  netmask 255.255.255.0
  gateway 192.168.1.1
```

Pevná IP adresa 192.168.1.2 je přiřazena síťovému rozhraní eth0 s maskou 255.255.255.0 a odchozí bránou 192.168.1.1.

INSTALACE PBX ASTERISK

Instalace softwarové pobočkové ústředny se skládá z několika částí. Asterisk není instalován přímo z balíčků, které poskytuje distribuce Debian, ale je instalován ze zdrojových kódu od výrobce a musí být kompilován. Proto se v průběhu instalace stahují balíčky, jenž jsou nutné pro kompilaci programu či jeho chod. Celá instalace je prováděna pod účtem **root**.

Příprava před instalací Asterisku

Před samotnou instalací Asterisku je potřeba provést stažení linuxového kernelu ve verzi 2.6 a příslušných balíčků, které zajistí kompilaci jádra.

První příkaz stáhne z repozitáře zdrojový kód jádra a k němu další důležité programy pro jeho kompilaci. Následuje rozbalení archívu do složky /usr/src. Dále se vytvoří symlink. Poté se provede nakopírování konfiguračního souboru a následuje souhrn příkazů pro kompilaci jádra. Postup pro stažení a kompilaci se skládá z těchto příkazů:

```
# aptitude install linux-source-2.6 kernel-package make g++
  libncurses5-dev
# cd /usr/src/
# tar xvjf /usr/src/linux-source-2.6.32.tar.bz2
# ln -s /usr/src/linux-source-2.6.32 /usr/src/linux
# cp /boot/config-`uname -r` /usr/src/linux/.config
# cd linux/
# yes "" | make oldconfig
# make prepare
# make prepare scripts
```

Dalším nezbytným krokem je instalace encoderu lame a jeho knihovny. Nejprve se musí přidat nový repositář do seznamu pro přidání balíčků.

```
# echo deb http://www.deb-multimedia.org squeeze main non-free >>
/etc/apt/sources.list
```

Následně je potřeba provést aktualizaci dostupných balíčků příkazem:

```
# aptitude update
```

Instalace **lame** a **libmp3lame-dev** se provede zadáním:

```
# aptitude install lame libmp3lame-dev
```

Instalace Asterisku

Asterisk je instalován ze zdrojových kódů. To znamená, že se musí provést kompilace a poté se nainstaluje. V průběhu instalace probíhá také nastavení parametrů ústředny. Prvním krokem je stažení a nainstalování potřebných balíčků, které jsou využity při kompilaci, instalaci a nastavení Asterisku. Získání balíčků se zajistí:

V následujícím kroku se provede stažení zdrojový kód Asterisku z Internetu.

Příkazem **wget** je z Internetu stažen aktuální balíček Asterisku, který je poté rozbalen do příslušné složky. Následně se spustí skript **get_mp3_source.sh** a ten stáhne podporu pro soubory MP3. Poté se spustí konfigurace, kdy proběhne kontrola jednotlivých souborů.

Příkaz **make menuconfig** zobrazí menu s výběrem jednotlivých modulů a nastavením kompilace. Z menu se vyberou položky z **Add-ons** → **format_mp3**, **app_mysql**, **app_saycountpl**, **res_config_mysql**, **cdr_mysql**. Dalé se v **Extras Sound Packages** označí položka **EXTRA-SOUNDS-EN-GSM**, což je zvuková sada pro Asterisk. V případě, že se kompilace bude provádět na novější hardwarové sestavě, může docházet při spuštění Asterisku k chybě. Pro správný chod je potřeba v oddíle **Compiler Flags** odznačit **BUILD_NATIVE**. Posloupnost příkazů je:

```
# cd /usr/src/
# wget http://downloads.asterisk.org/pub/telephony/asterisk/
old-releases/asterisk-1.8.20.1.tar.gz
# tar zxvf asterisk-1.8.20.1.tar.gz
# cd asterisk-1.8.20.1/
# ./contrib/scripts/get_mp3_source.sh
# ./configure
# make menuconfig
```

```
*****
Asterisk Module and Build Option Selection
*****

Press 'h' for help.

---> Add-ons (See README-addons.txt)
Applications
Bridging Modules
Call Detail Recording
Channel Event Logging
Channel Drivers
Codec Translators
Format Interpreters
Dialplan Functions
PBX Modules
Resource Modules
Test Modules
Compiler Flags
Voicemail Build Options
Utilities
AGI Samples
Module Embedding
Core Sound Packages
Music On Hold File Packages
Extras Sound Packages
```

Obr. 11. Výběr jednotlivých položek před kompilací

V dalším kroku se následuje kompilace a instalace Asterisku zadáním příkazů. Příkaz **make** zkompiluje kód a další **make install** provede instalaci Asterisku. Ukázkové nastavení se nainstaluje příkazem **make samples**.

```
# make
# make install
# make samples
```

INSTALACE FreePBX

Instalace grafického rozhraní FreePBX se skládá z několika důležitých částí. Zdrojové kódy FreePBX se získají z Internetu. Dále je nutné nainstalovat i další potřebné balíčky pro chod FreePBX. Verze FreePBX je 2.9.0., která se po instalaci zaktualizuje na nejnovější verzi (duben 2013) 2.10.0. Sled příkazů je:

```
# cd /usr/src/
# wget http://mirror.freepbx.org/freepbx-2.9.0.tar.gz
# tar zxvf freepbx-2.9.0.tar.gz
# aptitude install libxml2 libxml2-dev libtiff4
libtiff4-dev apache2 libapache2-mod-php5 php5-mysql
php5-gd php-pear mysql-server openssl libssl-dev
```

```
linux-source-2.6.32 bison libncurses5-dev
libaudiofile-dev curl sox
```

Při instalaci balíčku **mysql-server** je uživatel vyzván k zadání hesla, které využívá *root* MySQL uživatel.

V konfiguračním souboru `/etc/php5/apache2/php.ini` se provede změna parametrů u položek **upload_max_filesize** (povolení maximální velikost souborů pro upload) a **memory_limit** (limit paměti, které může využít skript). Příkazy pro snadnou úpravu jsou následující:

```
# sed -i "s/(upload_max_filesize *= *)\\(.*\\)/\120M/"
/etc/php5/apache2/php.ini
# sed -i "s/(memory_limit *= *)\\(.*\\)/\1100M/"
/etc/php5/apache2/php.ini
```

Nezbytným krokem je také vytvoření samostatného uživatele a skupiny, který bude sloužit pro chod FreePBX. Uživatel je pojmenován *asterisk* a skupina má opět stejné jméno *asterisk*. Vytvoření skupiny a uživatele se provede příkazy:

```
# groupadd asterisk
# useradd -c "Asterisk PBX" -d /var/lib/asterisk -g asterisk asterisk
```

V následující posloupnost příkazů vytváří databáze (*asteriskcdrdb*, *asterisk*) a uživatele *asteriskuser* s heslem *amp109* na MySQL serveru.

```
# mysql -u root -p
# Enter password:
# mysql> create database asteriskcdrdb;
# mysql> create database asterisk;
# mysql> GRANT ALL PRIVILEGES ON asteriskcdrdb.*
TO asteriskuser@localhost IDENTIFIED BY 'amp109';
# mysql> GRANT ALL PRIVILEGES ON asterisk.*
TO asteriskuser@localhost IDENTIFIED BY 'amp109';
# mysql> quit;
```

Vytvořené databáze pro běh FreePBX jsou zatím prázdné. Vložení příslušných příkazů se naplní databáze *asteriskcdrdb* a *asterisk*. Vkládání údajů do tabulek se provádí pod uživatelem *asteriskuser*.

```
# mysql -u asteriskuser -pamp109 asteriskcdrdb <
/usr/src/freepbx-2.9.0/SQL/cdr_mysql_table.sql
# mysql -u asteriskuser -pamp109 asterisk <
/usr/src/freepbx-2.9.0/SQL/newinstall.sql
```

Následující příkazy se zaměřují na již samotnou instalaci balíku FreePBX. Předchozí kroky se věnují přípravě, která musí být splněna před tím, než se provede instalace FreePBX. Nejdříve se vytvoří záloha souboru `/etc/asterisk/sip_notify.conf`. Příkazem **start_asterisk start** se spustí Asterisk. Je také nutné doinstalovat doplněk pro PHP, který vyžaduje FreePBX pro svůj chod.

```
# mv /etc/asterisk/sip_notify.conf /etc/asterisk/sip_notify.conf.backup
# /usr/src/freepbx-2.9.0/start_asterisk start
# pear install DB
# /usr/src/freepbx-2.9.0/install_amp
```

Spuštěním příkazu `/usr/src/freepbx-2.9.0/install_amp` se provede instalaci FreePBX. Instalátor je interaktivní a uživatele vybízí k zadávání údajů. Během instalace je uživatel vyzván k těmto událostem:

```
Enter your USERNAME to connect to the 'asterisk' database: [asteriskuser]
Enter your PASSWORD to connect to the 'asterisk' database: [amp109]
Enter the hostname of the 'asterisk' database: [localhost]
Enter a USERNAME to connect to the Asterisk Manager interface: [admin]
Enter a PASSWORD to connect to the Asterisk Manager interface: [amp111]
Enter the path to use for your AMP web root: [/var/www/html] /var/www
Enter the IP ADDRESS or hostname used to access the AMP
  web-admin: [192.168.1.1] 192.168.1.11
Enter a PASSWORD to perform call transfers with the
  Flash Operator Panel: [passwOrd]
Use simple Extensions [extensions] admin or separate Devices and
  Users [deviceanduser]? [extensions]
Enter directory in which to store AMP executable scripts:
  [/var/lib/asterisk/bin]
Enter directory in which to store super-user scripts:
  [/usr/local/sbin]
```

Všechny hodnoty jsou nastaveny jako výchozí kromě bodu pro zadání cesty a IP adresy pro přístup k webovému rozhraní.

Běh webového rozhraní má na starost webový server Apache. Pro případ nějaké chyby v konfiguraci se udělá záloha konfiguračního souboru `/etc/apache2/apache2.conf`. Poté je možné provést požadované změny. U Apache se nastaví přístupová práva pro uživatele a skupinu *asterisk* v daném souboru. Z důvodu provedených změn je potřeba zastavit Apache (příkaz **apache2 stop**) a poté ho znovu (příkaz **apache2 start**) spustit. Příkazy pro úpravu přístupových práv jsou následující:

```
# cp /etc/apache2/apache2.conf /etc/apache2/apache2.conf-orig
# sed -i "s/\(^User *\)\)\(.*)/\1asterisk/" /etc/apache2/apache2.conf
# sed -i "s/\(^Group *\)\)\(.*)/\1asterisk/" /etc/apache2/apache2.conf
# /etc/init.d/apache2 stop
# /etc/init.d/apache2 start
```

Poslední krokem instalace je vytvoření spouštěcího skriptu, který zajišťuje spuštění **amportalu**. To znamená, že Asterisk se spouští ihned po startu operačního systému. Skript je umístěn ve složce `/etc/init.d` a má název **amportal-startup**. Zdrojový kód je vložen v příloze. Příkaz pro vytvoření souboru otevře textový editor **nano**, do kterého se vloží příslušný zdrojový kód:

```
# nano /etc/init.d/amportal-startup
```

Vytvořenému souboru je potřeba přiřadit práva pro spuštění. To se provede příkazem:

```
# chmod +x /etc/init.d/amportal-startup
```

Skript je nutné umístit do tzv. init skriptů, které se spouštějí po startu operačního systému. Následující příkaz zajistí jeho přiřazení:

```
# inserv /etc/init.d/amportal-startup
```

Nyní se provede restart systému a po jeho načtení je možné přes webový prohlížeč přistoupit ke konfiguraci FreePBX.

```
# reboot
```

PŘÍLOHA P V. LABORATORNÍ ÚLOHA Č. 2 - ZABEZPEČENÍ ÚSTŘEDNY

Úkoly

1. Seznamte se zadáním a instalačním postupem u laboratorní úlohy č.2.
2. Nastavte a zprovozněte firewallový skript na OS Debian. Ověřte jeho nastavení.
3. Proveďte změnu výchozích hesel v konfiguračních souborech.
4. Proveďte změnu výchozích hesel ve webovém rozhraní FreePBX.

NASTAVENÍ A ZPROVOZNĚNÍ FIREWALLOVÉHO SKRIPTU

Zabezpečení firewallu

Firewallový skript má na starost filtrování povolených IP adres a služeb, jenž jsou poskytovány serverem. Pravidla skriptu jsou definována pomocí **iptables** a má název **firewall.sh**. Soubor s pravidly je uložen ve složce `/etc/init.d` a spouští se ihned po startu operačního systému. *Celý zdrojový kód skriptu je vložen příloze této laboratorní úlohy.*

Skript je rozdělen na dvě části. Část **start_firewall ()** slouží k zavedení nastavených pravidel. Spuštěním této funkce se nastaví všechna pravidla a přístup k serveru mají pouze povolené IP adresy a také jsou povoleny dané služby (porty). Funkce **stop_firewall ()** má na starost smazání všech vložených pravidel a nastavení původních hodnot **iptables**. To znamená, že k serveru má přístup jakákoliv IP adresa a může využít kterýkoliv port.

Firewall má povoleny porty pro služby SSH (TCP 22), SIP (TCP 5060, UDP 5060), DNS (UDP 53), HTTP (TCP 80), RTSP (TCP 554, UDP 554), FTP (TCP 21), FTP-DATA (TCP 20), IAX2 (UDP 4569) a přenos hlasu (UDP 10000 - 20000). Pouze přes tyto porty lze provádět komunikaci s okolím. Pro úspěšnou komunikaci s ústřednou je nutné znát port dané služby, ale také se musí přistupovat z povolené IP adresy.

Jelikož se skript spouští automaticky po startu operačního systému, musí být skript **firewall.sh** umístěn ve složce `/etc/init.d` a také je nutné vytvořit symbolické odkazy na firewallový skript do složek, které reprezentují jednotlivé úrovně běhu systému. Jsou to složky `/etc/rc0.d`, `/etc/rc1.d`, `/etc/rc2.d`, `/etc/rc3.d`, `/etc/rc4.d`, `/etc/rc5.d`, `/etc/rc6.d`. Skriptu se musí přiřadit práva pro spuštění. To se provede následovně:

```
# chmod +x /etc/init.d/firewall.sh
```

Vytvoření symbolických odkazů pro jednotlivé úrovně se provede příkazy:

```
# ln -s /etc/init.d/firewall.sh /etc/rc0.d/K98firewall.sh
```

```
# ln -s /etc/init.d/firewall.sh /etc/rc1.d/K98firewall.sh
# ln -s /etc/init.d/firewall.sh /etc/rc2.d/S98firewall.sh
# ln -s /etc/init.d/firewall.sh /etc/rc3.d/S98firewall.sh
# ln -s /etc/init.d/firewall.sh /etc/rc4.d/S98firewall.sh
# ln -s /etc/init.d/firewall.sh /etc/rc5.d/S98firewall.sh
# ln -s /etc/init.d/firewall.sh /etc/rc6.d/K98firewall.sh
```

Při vytvoření symbolických odkazů se vyskytnou písmena S a K. Písmeno S označuje spuštění a K ukončení služby. Posledním krokem k automatickému spouštění skriptu je aktualizace seznamu služeb, která se vykoná následujícím příkazem:

```
# update-rc.d firewall.sh defaults
```

Ověření nastavených pravidel se provede příkazem:

```
# iptables -vnL
```

Ovládání firewallového skriptu je také možné manuálně cestou. Skript je možné spustit, zastavit nebo restartovat. Spuštění nastaví všechna zadaná pravidla. Zastavení zajistí vymazání všech pravidel a nastavení původních hodnot. Restart ukončí a znovu spustí firewall.

Příkaz pro manuální spuštění:

```
# sh /etc/init.d/firewall.sh start
```

Příkaz pro manuální ukončení:

```
# sh /etc/init.d/firewall.sh stop
```

Příkaz pro manuální restart:

```
# sh /etc/init.d/firewall.sh restart
```

ZMĚNA VÝCHOZÍCH HESEL V KONFIGURAČNÍCH SOUBORECH

Instalace Asterisku a FreePBX se provádí s výchozími hesly. Jedním ze základních bezpečnostních prvků je změna a volba dostatečně silného hesla. Hesla se mění v konfiguračních souborech pro uživatele *asteriskuser*, který je používán pro MySQL. Změna se provede pod uživatelem *root* a zadáním následujících příkazů:

```

# NEWPASSWORD='heslo';
# sed -i "s/\(^AMPDBPASS=*\)\\(.*)\\/\1$NEWPASSWORD/"
    /etc/ampportal.conf
# sed -i "s/\(^password = *\)\\(.*)\\/\1$NEWPASSWORD/"
    /etc/asterisk/cdr_mysql.conf
# sed -i "s/\(^AMPDBPASS = *\)\\(.*)\\/\1$NEWPASSWORD/"
    /etc/asterisk/extensions_additional.conf
# sed -i "s/\($amp_conf\[ 'AMPDBPASS' \] \\t= '*\)\\(.*)\\/\
1$NEWPASSWORD';
/" /etc/freepbx.conf
# echo "SET PASSWORD FOR 'asteriskuser'@'localhost' =
    PASSWORD('$NEWPASSWORD');" | mysql -u root -p

```

Další krokem je změna výchozích hesel u Asterisk Manageru. Příkazy pro provedení změny jsou:

```

# NEWPASSWORD='heslo';
# sed -i "s/\(^secret = *\)\\(.*)\\/\1$NEWPASSWORD/"
    /etc/asterisk/manager.conf
# sed -i "s/\(^AMPMGRPASS = *\)\\(.*)\\/\1$NEWPASSWORD/"
    /etc/asterisk/extensions_additional.conf
# sed -i "s/\(^AMPMGRPASS=*\)\\(.*)\\/\1$NEWPASSWORD/"
    /etc/ampportal.conf
# asterisk -r -x "module reload manager"
# echo "UPDATE freepbx_settings SET value='$NEWPASSWORD'
    WHERE keyword='AMPMGRPASS';" | mysql -u root -p asterisk

```

ZMĚNA VÝCHOZÍCH HESEL VE WEBOVÉM ROZHRANÍ FREEPBX

Nyní jsou změněny všechna hesla, která se používala při instalaci. Na řadě je tedy změna hesel, která se týká samotného FreePBX. Změna hesla administrátora se mění v **Admin** → **Administrators**, kde se v pravé části vybere položka **admin**.

Ostatní hesla související s provozem FreePBX se mění v sekci **Settings** → **Advanced Settings**. Změna hesel je nutná u **FOP Password** a **User Portal Admin Password**.

PŘÍLOHA P VI. LABORATORNÍ ÚLOHA Č. 3 - NASTAVENÍ ÚSTŘEDNY

Úkoly

1. Seznamte se zadáním a instalačním postupem u laboratorní úlohy č.3.
2. Přihlaste se do FreePBX a proveďte aplikování změn po instalaci.
3. Proveďte aktualizaci FreePBX na nejnovější verzi.
4. Nainstalujte přídatné moduly pro ovládání FreePBX.
5. Vytvořte klapku typu SIP.
6. Vytvořte IAX trunk k poskytovateli hlasových služeb. Nastavte odchozí a příchozí cesty.
7. Vytvořte IAX trunk ke školní ústředně a nastavte odchozí cestu.

PŘIHLÁŠENÍ DO FREEPBX A APLIKOVÁNÍ ZMĚN

Základní nastavení

Veškeré nastavení ústředny se provádí přes webové rozhraní FreePBX. Do webového prohlížeče se vloží IP adresa serveru např. <http://192.168.1.11/admin>. Prozatím je nastaveno výchozí uživatelské jméno a heslo pro administrátora.

Uživatelské jméno: admin

Heslo: admin

Po instalaci je také potřeba provést aplikování změn ve FreePBX. Kliknutím na oranžový pásek s textem ***Apply Configuration Changes*** a potvrzením dialogu ***Continue with reload*** se nastavení uloží do systému.

AKTUALIZACE FREEPBX NA NEJNOVĚJŠÍ VERZI

Nainstalovaná verze FreePBX je ve verzi 2.9.0.7. Jelikož je již vydána verze 2.10.0, provede se aktualizace na nejnovější stabilní verzi. V levém sloupci se přejde na ***Module Admin*** a poté se klikne ***Check Online***. Objeví se tabulka vybízející k aktualizaci jednotlivých modulů. Celková aktualizace zajistí instalaci modulu ***2.10 Upgrade Tool***. Pro instalaci se vybere sekce ***Action*** a položka ***Download and Install***. Stiskem tlačítka ***Process*** a potvrzením ***Confirm*** se provede instalace modulu.

V levém panelu přibyla nová sekce ***2.10 Upgrade Tool***. Po přístupu na ni se klikne na tlačítko ***UPGRADE NOW*** a potvrdí se stiskem ***OK***. Opět je nutné se vrátit do sekce ***Module Admin***, kde se provede aktualizace. Stisknutím tlačítka ***Check Online*** a výběru položky ***FreePBX Framework*** a kliknutím na ***Download and***

Upgrade to 2.10.1.9 se určí, co se má aktualizovat. Poté je postup stejný jako v případě instalace aktualizacího modulu. FreePBX začne stahovat z Internetu novou verzi a poté ji již nainstaluje do systému.

Aktualizace ještě není kompletní. Je nutné se zase vrátit do sekce **Module Admin** a zde znovu provést kontrolu jednotlivých modulů. Aktualizují se moduly **FreePBX ARI Framework**, **FreePBX Framework** a **Core**. Opětovným načtením stránky se klikne na **Apply Config**. Nová verze FreePBX s sebou přináší změnu grafického vzhledu.

Pro celkové dokončení instalace se musí zaktualizovat moduly v sekci **Admin** → **Module Admin**. Jedná se o **Custom Applications**, **Feature Code Admin**, **FreePBX FOP Framework**, **Recordings**, **Info Services**, **System Dashboard**, **Music on Hold** a **Voicemail**. Postup aktualizace je opět stejný. Provedené změny je nutné uložit stiskem červeného tlačítka **Apply Config**. Nyní je aktualizace kompletní.

VYTVOŘENÍ KLAPKY TYPU SIP

Přidání nové klapky se ve FreePBX řeší v oddíle **Applications**. Jelikož se v této práci pracuje s protokolem SIP, vybere se typ zařízení **Generic SIP Device**. Zobrazí se formulář pro přidání nového SIPového zařízení.

Pro vytvoření nové klapky je potřeba vyplnit základní formulářová pole. Prvním polem je **User Extension**, do kterého se vloží číslo klapky (např. 101). Do **Display Name** se napíše jméno uživatele (např. Jan Novák), jenž je zobrazeno při volání. Položka **Outbound CID (CallerID)** obsahuje telefonní číslo, které se zobrazí při volání do veřejné sítě přes odchozí cestu a musí být vloženo, jinak hovor neproběhne. SIPové zařízení je chráněno heslem. Heslo se vkládá do pole **secret**, kde se napíše definované heslo. Obsahovat může alfanumerické znaky. V případě, že vytvořená klapka je umístěna za NATem, musí se vybrat u položky **nat** parametr **yes**. Pokud by se tak neudělalo, klapka by nebyla přístupná pro příchozí volání z Internetu. Ústředna již má v sobě implementovanou hlasovou schránku. Aktivace hlasové schránky se provedete výběrem volby **Enabled** u položky **Status**. Přístup do hlasové schránky je chráněn heslem. Heslo se zadává pouze jako čísla a vkládá se do pole **Voicemail Password**. Všechny provedené změny se uloží stiskem tlačítka **Submit**.

VYTVOŘENÍ IAX TRUNKU K POSKYTOVATELI HLASOVÝCH SLUŽEB

Základní nastavení

K poskytovateli je vytvořen IAX trunk, který zaslal přihlašovací parametry (viz níže) pro připojení k jeho ústředně a samozřejmě také poskytl veřejná telefonní čísla pro

volání.

Vytvoření trunku se provede přes položky *Connectivity* → *Trunks* a zvolí se *Add IAX2 Trunk*. V sekci *General Settings* se do pole *Trunk Name* napíše jméno trunku, které by mělo být výstižné (např. IAXtrunk_to_provider). Oddíl *Outgoing Settings* je nejdůležitější částí. Obsahuje údaje pro nastavení trunku. Zde je také opět pole *Trunk Name*, do které se napíše unikátní jméno trunku (např. IAX_trunk). Položka *PEER Details* obsahuje následující parametry:

```
username=uziv_iax
type=friend
secret=heslo
host=sip.poskytovatel.cz
requirecalltoken=no
qualify=yes
context=from-trunk
```

Položka *username* označuje uživatelské jméno a *secret* má přiřazeno heslo pro autentizaci na straně poskytovatele. Řádek *type* vyjadřuje vztah mezi školní ústřednou a vzdálenou ústřednou, který je typu *friend*. Adresa ústředny poskytovatele je zadána v položce *host*. U *requirecalltoken=no* není vyžadována Call Token Validation. Parametr *qualify=yes* zajišťuje kontrolu spojení mezi školní ústřednou a poskytovatelem. Pole *context* určuje, že se jedná o spojení z trunku.

Oddíl *Incoming Settings* slouží pro ověření příchozí komunikace a má pojmenovaný *USER Context* *uziv_iax*. V *USER Details* jsou vloženy parametry:

```
secret=heslo
type=user
context=from-trunk
```

Položka *secret* obsahuje heslo, které je stejné jako v nastavení výše. Parametr *type=user* vyjadřuje vztah a *context=from-trunk* je označení spojení.

Další polem, které je vyplněno v oddíle *Registration*, je položka *Register String* má registrační řetězec:

```
uziv_iax:heslo@sip.poskytovatel.cz
```

Tento registrační řetězec registruje trunk u poskytovatele. Stiskem tlačítka *Submit Changes* se uloží nastavení.

Nastavení odchozí cesty (Outbound Routes)

Odchozí cesta zajišťuje odeslání hovoru na vytvořený trunk k poskytovateli. Obsahuje vytvořené prefixy, které jsou povoleny pro danou odchozí cestu.

Route Name obsahuje pojmenování dané cesty. Srdcem celé routy jsou vzory předvoleb (**Dial Patterns**), které právě směřují vzory na trunk k poskytovateli. Jsou vytvořeny dvě odchozí cesty. Každá plní konkrétní úlohu. Je vytvořena cesta pro *Emergency* (tísňová volání), *Vnitrostatni* (volání v rámci státu).

Vzory předvoleb pro *Emergency* obsahují tísňová volání, která jsou používána v České republice. Jsou to čísla 112, 150, 155, 156 a 158. U položky **Route Type** je zaškrtnuto **Emergency**, jenž označuje tísňová volání.

() + | [112 /]

() + | [15[0568] /]

Vnitrostatni obsahuje vzory pro všechna devítimístná národní čísla a pro vybraná čísla jako např. 14112 pro zjištění aktuálního času.

() + | [14112 /]

() + | [NXXXXXXXX /]

() + | [00. /]

Nastavení příchozí cesty (Inbound Routes)

Nastavení příchozí cesty se nastaví přes **Connectivity** -> **Inbound Routes**. Zobrazí se formulář pro přidání. Do pole **DID Number** se vloží přiřazené veřejné telefonní číslo. Nyní se musí toto číslo spárovat s danou klapkou. Toto se děje v položce **Set Destination**, kde se vybere **Extensions** a příslušná klapka. Kliknutím na **Submit** se uloží nastavení. Nyní je příchozí hovor z určeného veřejného čísla přeměrován na vybranou klapku.

VYTVOŘENÍ IAX TRUNKU KE ŠKOLNÍ ÚSTŘEDNĚ

Vytvoření IAX trunku ke školní ústředně

Vytvoření trunku je stejné jako v případě k poskytovateli hlasových služeb. Změna je pouze v parametrech. Položka **Trunk Name** pojmenovává trunk. V oddíle **Outgoing Settings** je **Trunk Name** pojmenován *skolni-peer*. **PEER Details** obsahuje parametry:

username=student-user

type=peer

trunk=yes

secret=heslo

qualify=yes

host=sip.fai.utb.cz

Oddíl *Incoming Settings* má *USER Context* pojmenování *skolni-user*. Sekce *USER Details* obsahuje:

```
secret=heslo  
type=user  
context=from-trunk
```

Položka `type=user` vyjadřuje, že se jedná o IAX2 spojení. Uložení nastavení se provede stiskem tlačítka *Submit Changes*.

Nastavení odchozí cesty ke školní ústředně

Odchozí cesta má opět pojmenování v položce *Route Name*. Vzor říká, že všechna odchozí čísla 1XX, jsou směřována do trunku. To znamená, že je vytočeno číslo 01XX, kde 0 značí odchozí hovor z ústředny.

```
( ) + 0 | [1XX / ]
```

V sekci *Trunk Sequence for Matched Routes* se vybere příslušný odchozí trunk na školní ústřednu. Uložení se provede stiskem *Submit Changes*.