



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

REALIZACE APLIKACE TYPU AUTOMATIC ATTENDANT POMOCÍ ÚSTŘEDNY ASTERISK

AUTOMATIC ATTENDANT APPLICATION USING ASTERISK PBX

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

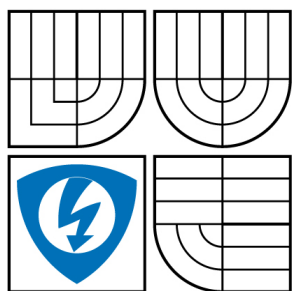
JIŘÍ BENÝŠEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PETR KOVÁŘ

BRNO 2008



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor

Teleinformatika

Student: Benýšek Jiří

ID: 78180

Ročník: 3

Akademický rok: 2007/2008

NÁZEV TÉMATU:

Realizace aplikace typu Automatic Attendant pomocí ústředny Asterisk

POKYNY PRO VYPRACOVÁNÍ:

Analyzujte potřeby hypotetické společnosti s cca dvaceti pobočkami a call centrem o 4 pobočkách. Stanovte klíčové požadavky na aplikaci typu Automatic Attendant, nasazené v těchto podmínkách. Prostudujte open source ústřednu Asterisk.

Na základě závěrů získaných v teoretické části realizujte pomocí ústředny Asterisk aplikaci automatic attendant pro výše zmíněnou hypotetickou společnost. Diskutujte vhodnost využití ústředny Asterisk pro tento úkol.

Jako přílohu práce vypracujte laboratorní úlohu, která bude prakticky demonstrovat dosažené závěry.

DOPORUČENÁ LITERATURA:

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

[2] GOMILLION, D., DEMPSTER, B., Building Telephony Systems with Asterisk. Packt Publishing, 2005. ISBN 978-1904811152

Termín zadání: 11.2.2008

Termín odevzdání: 4.6.2008

Vedoucí práce: Ing. Petr Kovář

prof. Ing. Kamil Vrba, CSc.

předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

LICENČNÍ SMLOUVA

POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Jiří Benýšek
Bytem: Hlušovice 83, 78314, p. Bohunovice
Narozen/a (datum a místo): 15.11.1985, Šternberk

(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 244/53, 60200 Brno 2
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Ing. Kamil Vrba, CSc.

(dále jen "nabyvatel")

Článek 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- disertační práce
- diplomová práce
- bakalářská práce

jiná práce, jejíž druh je specifikován jako

(dále jen VŠKP nebo dílo)

Název VŠKP: Realizace aplikace typu Automatic Attendant pomocí ústředny
Asterisk

Vedoucí/školitel VŠKP: Ing. Petr Kovář

Ústav: Ústav telekomunikací

Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

- tištěné formě - počet exemplářů 1
- elektronické formě - počet exemplářů 1

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.

3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.

4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....

Nabyvatel

.....

Autor

ANOTACE

V současné době je IP telefonie na velkém vzestupu. Vedle profesionálních řešení určených pro velké společnosti se prosazuje také do malých organizací a domácností. Nabízené ústředny většinou obsahují pouze základní funkce a pokud chce zákazník využít pokročilých možností, musí sáhnout k nákladnému řešení. Tento stav se ale začíná pomalu měnit. Na trh přicházejí open source řešení, jejichž tvůrci proklamují, že jejich projekt dokáže nahradit i plně profesionální produkty. Jedním z nich je i pobočková ústředna Asterisk.

Bakalářská práce je zaměřena na problematiku IP telefonie, pojednává o základních principech a protokolech využívaných touto službou. Dále pojednává o softwarové ústředně Asterisk a jejích možnostech pro zabezpečení funkcí požadovaných provozem v call centrech, zejména ověřením možností vytvoření a správy volacích front. Po úvodním teoretickém rozboru se práce zabývá instalací a konfigurací ústředny. To se týká zejména instalace potřebného operačního systému, doplňujících balíčků obsahujících nutné knihovny a moduly pro její správnou funkci ústředny. Další část se věnuje konfiguraci Asterisku, zejména pak vytvořením a nastavením uživatelských účtů protokolu SIP a vytvoření číslovacího plánu. Následuje ověření možností vytvoření a spravování volacích front a to zejména s přihlédnutím k nastavení jednotlivých parametrů, připojováním operátorů a zařazením do číslovacího plánu. Poslední kapitola bakalářské práce prezentuje dosažené praktické výsledky.

Klíčová slova: IP telefonie, SIP, PBX, Asterisk, call centrum, fronta, člen, Linux

ABSTRACT

Application of IP telephony is still more and more popular. This technology is designed for large companies as well as for home or office use. If customer needs to use advanced features of the private branch exchanges, he had to purchase more expensive solutions than common equipment. These professional products may be replaced by open source solutions, which can fully replace them. Private branch exchange Asterisk is one example of these products.

Bachelor's thesis is focused on IP telephony, treats of basic principles and network protocols used by this service. This thesis then describes the public branch exchange Asterisk, its possibilities for use in call centers and administration of call queues using this PBX. After theoretical analysis, this thesis deals with installation and configuration of this PBX. It includes installation of necessary operating system, additional packages containing necessary libraries and modules needed for correct function of the PBX. Configuration of Asterisk such as creation of user accounts of SIP protocol and dialplan is described in the next chapter and then all of these functions are tested. Achieved practical results are presented in the last chapter of this bachelor's thesis.

Keywords: IP telephony, SIP, PBX, Asterisk, call centrum, queue, member, Linux

BIBLIOGRAFICKÁ CITACE MÉ PRÁCE

BENÝŠEK, J. *Realizace aplikace typu Automatic Attendant pomocí ústředny Asterisk*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 49 s. Vedoucí bakalářské práce Ing. Petr Kovář.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Realizace aplikace typu Automatic Attendant pomocí ústředny Asterisk“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

V Brně dne

.....
podpis autora

PODĚKOVÁNÍ

Rád bych na tomto místě poděkoval mému vedoucímu bakalářské práce Ing. Petru Kováři, za velmi užitečnou metodickou pomoc, rady při zpracování tohoto dokumentu a za čas, který věnoval mé bakalářské práci.

V Brně dne

.....
podpis autora

SEZNAM POUŽITÝCH ZKRATEK

ACD	Automatic Call Distributor
ADSL	Asymmetric Digital Subscriber Line
BSD	Berkeley Software Distribution
CLI	Command Line Interface
DNS	Domain Name System
DTMF	Dual-tone multi-frequency
GCC	GNU Compiler Collection
GNU	GNU's Not Unix
GSM	Global System of Mobile Communications
HTTP	Hypertext Transfer Protocol
IAX	Inter Asterisk Exchange
IETF	Internet Engineering Task
iLBC	internet Low Bitrate Codec
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ITU	International Telecommunication Union
IVR	Interactive Voice Response
LAN	Local Area Network
MC	Multipoint Controller
MCU	Multipoint Control Unit
MP	Multipoint Processor
MP3	MPEG-1 Layer 3
MPEG-1	Moving Picture Experts Group Phase 1
PBX	Private Branch Exchange
PSTN	Public Switched Telephone Network
RAM	Random Access Memory
RTP	Real-time Transport Protocol
RTCP	Real-time Transport Control Protocol
SIP	Session Initiation Protocol
SMTP	Simple Mail Transfer Protocol
TCP	Transmission Control Protocol
TDM	Time Division Multiplex
UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VOIP	Voice over Internet Protocol
WAV	Waveform

OBSAH

Úvod	13
1 IP telefonie	14
1.1 Co je IP telefonie	14
1.2 Základní princip IP telefonie	14
2 Používané protokoly v IP telefonii	16
2.1 Co je protokol	16
2.2 H.323	16
2.2.1 Architektura H.323	16
2.2.2 Sestavení spojení podle H.323	17
2.3 Protokol SIP	18
2.3.1 Architektura SIP	18
2.3.2 Žádosti a odpovědi	19
2.3.3 Identifikace volaného v SIP	20
2.3.4 Sestavení spojení podle SIP	20
3 Softwarová ústředna Asterisk	21
3.1 Popis PBX Asterisk	21
3.2 Nároky na hardware	22
3.3 Instalace potřebných částí	22
3.3.1 Linux	22
3.3.2 Používané balíčky	23
3.3.3 Zaptel	23
3.3.4 Ztdummy	23
3.3.5 Asterisk	24
3.3.6 Použité příkazy	24
3.4 Call centrum	25
3.4.1 Automatické rozdělování hovorů	25
3.4.2 Volací fronty	26
3.4.3 Hudba na pozadí	28
3.4.4 Hlasová samoobsluha	29
3.5 Konfigurace PXB Asterisk	30
3.5.1 Agents.conf	30
3.5.2 Sip.conf	31
3.5.3 Musiconhold.conf	31
3.5.4 Queues.conf	32
3.5.5 Extensions.conf	34
3.6 Spouštění Asterisku	35
3.6.1 Konzolový mód CLI	35
4 Softwarový klient X-Lite	36
4.1 Popis X-Lite	36
4.2 Základní nastavení	36

5 Praktické odzkoušení	38
5.1 Testovací zapojení.....	38
5.2 Konfigurace ústředny	39
5.3 Použité technické vybavení.....	39
Závěr.....	40
Literatura	41
Seznam příloh.....	42

SEZNAM OBRÁZKŮ

Obr.1: Průběh spojení podle H.323	17
Obr.2: Propojení UA se servery	19
Obr.3: Průběh spojení podle SIP	20
Obr.4: Popis základních funkcí X-Lite.....	37
Obr.5: Testovací zapojení.....	38

ÚVOD

Komunikace prostřednictvím IP telefonie je čím dál více aktuální. Tento fakt je nejspíše zapříčiněn stále narůstající kvalitou a dostupností širokopásmového Internetového připojení. Konkurenční boj nutí Internetové poskytovatele bojovat nejen cenou, ale i zkvalitňovat a rozšiřovat portfolio služeb. IP telefonie není však jen provázána s komunikací prostřednictvím Internetu. Jedná se o daleko komplexnější řešení zastřešující hlasové služby fungující obecně na každé síti, která využívá protokoly UDP/TCP/IP, přičemž koncové zařízení nemusí být jen počítač, ale třeba i IP telefon atd.

Ke spojování a řízení hovorů slouží analogicky jako u klasické telefonní sítě i v IP telefonii takzvaná telefonní ústředna. V našem případě jde o ústřednu 5. generace, kde je spojovací pole realizováno softwarově například pomocí VoIP. Nejčastěji používanou VoIP ústřednou v současnosti je Asterisk. Tato ústředna je komplexní systém umožňující realizovat spoustu VoIP řešení. Nespornou výhodou je i fakt, že tento systém je možno nainstalovat na běžný počítač bez dalšího hardware. V případě potřeby užití Asterisku v analogové telefonní síti, je možné ústřednu rozšířit o patřičný hardware.

1 IP TELEFONIE

1.1 Co je IP telefonie

IP telefonie neboli VoIP (Voice over Internet Protocol), je alternativa ke klasickému telefonování však s výrazně ekonomičtějšími nároky. Například za dálkový hovor uskutečněný touto službou pomocí sítě Internet třeba i na druhou stranu zeměkoule nezaplatíme v zásadě téměř nic (přesněji jen to, co normálně zaplatíte za přístup k Internetu).

Pod pojmem IP telefonie se tedy rozumí přenos hlasu po datových sítích. První standardy vznikly již v roce 1996 a dále se postupně vyvíjely. Hlasová informace je přenášena pomocí komunikačních sítí založených na IP protokolu v jedné síti společně s dalšími datovými informacemi jako jsou např. e-maily. Hovor zprostředkovaný pomocí IP telefonie se jeví pro koncového uživatele stejný jako hovor v současné telekomunikační síti. Výhodou tohoto řešení je, že IP telefonie může využít např. v rámci podniku k uskutečnění hovoru již existující datovou síť a plně se do její infrastruktury integrovat. Lze tedy konstatovat, že VoIP technologii je možné uplatnit obecně všude tam, kde lze provozovat IP protokol.

1.2 Základní princip IP telefonie

Analogový záznam hlasu je pro přenos po IP síti nepoužitelný, proto je třeba přenášený signál nejprve převést do číslicového tvaru. Zařízení, které převádí analogový signál, se označuje jako kodek, anglicky codec, což je zkratka pro COder/DECOder. Jde v podstatě o opak modemu, který na základě posloupnosti číslicových dat generuje spojitý analogový signál. Kodek na základě analogového signálu generuje posloupnost číslicových dat.

Podle Shannonova teorému o rekonstrukci spojitého, frekvenčně omezeného signálu z jeho vzorků vyplývá, že vzorkovací frekvence musí být alespoň dvakrát větší, než je maximální frekvence rekonstruovaného signálu. Vycházíme tedy ze vztahu:

$$f_{vz} \geq 2f_{max} \quad \text{kde: } f_{vz} \text{ je vzorkovací frekvence a } f_{max} \text{ je maximální frekvence}$$

Když by nastal případ, že by byla vzorkovací frekvence menší než maximální přenášená, mohlo by dojít k tzv. aliasingu (dochází k překrytí frekvenčních spekter vzorkovaného signálu a tedy ke ztrátě informace). V praxi to znamená, že u klasické telefonní sítě mající šířku pásma od 0,3 do 3,4kHz (maximální frekvence je zaokrouhlena směrem nahoru na 4kHz, neboť je ještě třeba počítat s rezervou) volíme vzorkovací frekvenci 8kHz. Kodek tedy generuje celkem 8000x8bitu, což je přesně kapacita jednoho digitálního okruhu (tj. 64kbit/s). Lidské sluchové ústrojí však není dokonalé, čehož v tomto případě lze využít v náš prospěch za použití tzv. ztrátových kompresních algoritmů. To jsou programy, které

využívají nedokonalosti sluchu (např. jevu zvaného maskování – u slabších signálů může dojít k přehlušení signálem s větší dynamikou) a na jejich základě vypouští redundantní data z nesené informace. Z poměrně dost vysokého základního datového toku 64kbit/s se pak můžeme dostat při zachování solidní kvality přenášeného hlasu až na hodnotu kolem 10kbit/s.

V IP telefonii potřebujeme nejen přenášet vlastní hovorová data, ale také potřebujeme navazovat spojení a přenášet další řídicí informace. Vlastní data se většinou přenášejí přes protokol RTP. Řídicí protokoly se v zásadě používají dva. Je to jednak sada protokolů H.323 a pak protokol SIP.

H.323 - vytvořený v rámci International Telecommunication Union (ITU). Tato sada v sobě zahrnuje více než 25 protokolů pro specifické aplikace. V rámci H.323 lze najít protokoly nabízející podporu pro sdílení dat, videokonference, audio aplikace v reálném čase, ale také protokoly pro navazování spojení, zabezpečení spojení a spoustu dalších.

Session Initiation Protocol (SIP) - vytvořený v rámci Internet Engineering Task Force (IETF) je alternativou ke komplexnímu H.323. Byl primárně vytvořen pro IP telefonii. Je výrazně jednodušší než H.323. SIP nabízí řadu služeb jako např. identifikaci volajícího, automatické směrování hovorů atd. SIP je oproti H.323 textový, a proto je snadněji analyzovatelný člověkem.

Real-time Transport Protocol (RTP) - vytvořený v rámci IETF. Popisuje standardní paketový formát pro doručování multimediálních dat po Internetu. Protokol RTP je přenášen pomocí UDP. Používá se ve spojení s H.323, SIP a společně s nimi tvoří základ VoIP. K řízení RTP relace se používá Real-time Transport Control Protocol (RTCP).

2 POUŽÍVANÉ PROTOKOLY V IP TELEFONII

2.1 Co je protokol

Protokol je soubor syntaktických a sémantických pravidel určujících výměnu informace mezi nejméně dvěma entitami spojenými například prostřednictvím počítačové sítě. Zahrnuje tyto procedury:

- proceduru navázání spojení
- přidělování prostředků
- adresování
- řízení toku komunikace
- přenos dat
- zpracování chyb

2.2 H.323

Protokol H.323 není vlastně samostatný protokol, ale sada protokolů pro přenos multimedií - tedy audia i videa. Tvoří ji následující doporučení:

- H.323 - popis architektury systému
- H.225.0 - signalizace
- H.245 - řízení volání
- H.235 - bezpečnost a šifrování
- H.332 - vytváření větších konferencí
- H.450 - doplňkové služby

2.2.1 Architektura H.323

H.323 definuje čtyři základní komponenty:

Terminal (terminál) - je základní a jedinou povinnou složkou sítě. Musí podporovat audio služby a případně může podporovat i video nebo datové služby. Komunikuje s ostatními terminály v síti. Terminál může být jak hardwarové, tak softwarové zařízení.

Gateway (brána) - slouží pro přenos multimediálních dat mezi různými systémy s odlišným protokolovým vybavením (např. H.320, H.324, ...). Slouží jako překladač protokolů. Překlad není nutný jen tehdy, když obě sítě využívají stejné protokoly. Je to volitelná komponenta.

MCU (více bodové řídicí jednotky) - zajišťuje skupinovou komunikaci mezi třemi a více terminály (konferenční hovory). MCU dělíme na dvě části a to:

- Multipoint controller (MC) - zajišťuje způsob komunikace a je to povinná část
- Multipoint processor (MP) - řeší tzv. mixování (přepínání videa, zvuku a dat)

Obě tyto části mohou pracovat dohromady nebo každá samostatně nezávisle na sobě. Takto řešené konference mohou být centralizované, decentralizované nebo hybridní.

Gatekeeper (správce) - v síti může být maximálně jeden. Na starost má autorizaci, autentifikaci, překlad telefonních čísel na IP adresy, účtování služeb a podobně. Jeho služby rozdělujeme do dvou skupin. První je povinná a druhá volitelná.

Povinné:

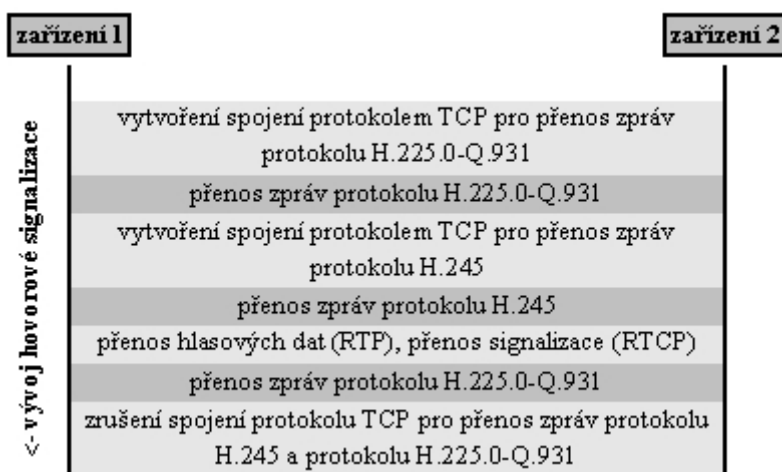
- Překlad adres podle E.164
- Kontrola pásma
- Řízení přístupu
- Řízení zóny

Volitelné:

- Autorizace hovoru
- Řízení signalizace
- Řízení hovorů
- Řízení pásma

2.2.2 Sestavení spojení podle H.323

Na Obr.1 je popsán časový sled událostí nastávajících při hlasovém spojení podle protokolů sady H.323. Spojení je popsáno pro dvě koncová zařízení. U H.323 je obecně nejprve zapotřebí vytvořit spojení prostřednictvím protokolu TCP. Přenos hlasových dat probíhá protokolem RTP (RTCP zajišťuje signalizaci) pomocí protokolu UDP.



Obr.1: Průběh spojení podle H.323

2.3 Protokol SIP

V dnešní době je více nasazován protokol SIP oproti staršímu H.323. Je používán jako signalizační protokol. K vlastnímu přenosu používá jiné existující standardy. Hojně je využit v IP telefonii, kde má na starost sestavení, modifikaci a ukončení spojení mezi dvěma a více uživateli. Spojení může ale obecně představovat i jakýkoliv jiný multimediální přenos, ne jen IP telefonii. SIP protokol je nedílnou součástí architektury TCP/IP. Za normálních podmínek používá UDP, ale funguje i na TCP. Protokol SIP je založen na textové bázi. Využívá principy známé z Internetových protokolů jako jsou HTTP a nebo poštovní protokol SMTP. Je velmi flexibilní, umožňuje jednoduché rozšiřování. Tento protokol pracuje na principu klient – server. Při této komunikaci dochází k výměně dvou typů zpráv (metoda – odpověď).

2.3.1 Architektura SIP

SIP architektura v rámci které funguje se skládá z následujících komponent [8]:

User Agent (UA) – je koncové zařízení (nejčastěji se jedná o SIP telefony nebo brány do jiných sítí). Má na starost navazování spojení s ostatními UA. Část UA, která zprostředkovává iniciaci spojení nazýváme User Agent Client (UAC). Naproti tomu část UA reagující na příchozí žádosti a posílající odpovědi označujeme jako User Agent Server (UAS). SIP telefony (softwarové i hardwarové) musí samozřejmě zahrnovat UAC i UAS.

Servery - zařízení mající na starost zprostředkování kontaktu mezi jednotlivými UA (mezi volajícím a volaným). V SIP architektuře rozlišujeme tři typy serverů:

Proxy server

Tento typ serveru přijímá žádosti o spojení od jednotlivých UA nebo od jiného proxy serveru. Pokud je volaný UA v rámci jeho správy, předá mu žádost. V případě, že volaná stanice není v rámci jeho správy, posílá ji na další proxy server.

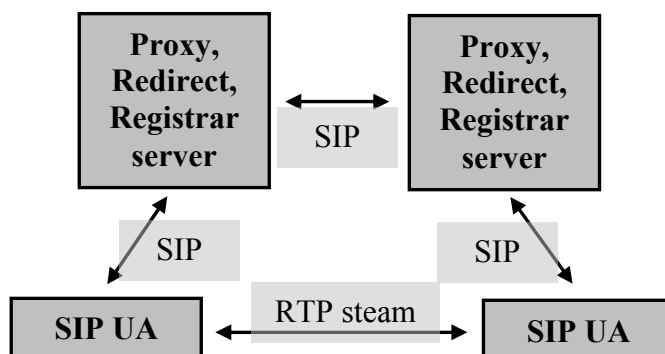
Redirect server

Opět dostává žádosti o spojení od UA nebo proxy serverů. Rozdíl od proxy serveru je v tom, že přijaté žádosti neposílá dál ve směru volané stanice, ale zpět k dotazujícímu s informací kam žádost poslat, aby se dostala k volanému. Dotazující potom musí poslat žádost na tuto získanou lokalitu, kde daného UA nebo proxy/redirect server hledat.

Registrar server

Zpracovává registrační žádosti od jednotlivých UA a podle nich pak aktualizuje databázi koncových zařízení (location service), které jsou spravovány v rámci domény.

I když jsou jednotlivé servery definovány odděleně, v praxi se mnohdy setkáváme s tím, že jde o jednu aplikaci. Tato aplikace pak přijímá registrace koncových uzlů a podle jejich konfigurace se chová zároveň buď jako proxy nebo redirect server.



Obr.2: Propojení UA se servery

2.3.2 Žádosti a odpovědi

V architektuře protokolu SIP je komunikace mezi komponenty řízena agenty a servery. Posílají si navzájem požadavky pomocí takzvaných žádostí. Jsou to zprávy v textovém formátu. Mezi základních patří [8]:

- INVITE - žádost o sestavení spojení
- ACK - potvrzení INVITE finálním příjemcem zprávy
- BYE - ukončení spojení
- CANCEL - ukončení nesestaveného spojení
- REGISTER - registrace UA
- OPTIONS - dotaz na možnosti a schopnosti serveru

Odpovědi na SIP žádosti jsou zprávy uvedené číselným kódem. Systém značení těchto kódů je převzat z HTTP protokolu:

- 1XX - informační zprávy
- 2XX - ukončení žádosti
- 3XX - přesměrování, dotaz je třeba směřovat jinam
- 4XX - chyba, dotaz je třeba zadat v jiné podobě
- 5XX - chyba na serveru
- 6XX - globální selhání

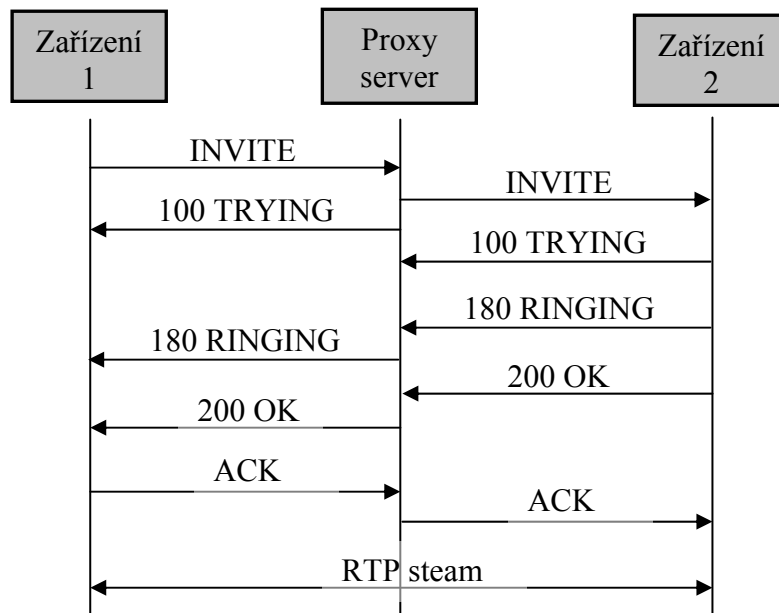
2.3.3 Identifikace volaného v SIP

U klasické telefonní sítě slouží pro identifikaci jednotlivých účastníků telefonní čísla. Protokol SIP naproti tomu používá k identifikaci účastníků již existující standardy URI a URL (popis zdrojů nacházejících na Internetu). Takovým způsobem lze identifikovat nejen koncové účastníky ale i brány do jiných sítí, hlasové záznamníky a spoustu dalších zařízení. Pár příkladů zápisů adres, které v rámci SIP protokolu můžeme vytáčet [8]:

- sip:654321@gw2pstn.abcd.com, user=phone
- sip:petr.novak@firma.cz
- sip: "Hlava Marek" marek@redmeat.com
- sip:123456@dalsifirma.cz
- tel:+420244470645

2.3.4 Sestavení spojení podle SIP

Na Obr.3 vidíme spojení mezi dvěma účastníky, kdy volající (zařízení 1) i volaný (zařízení 2) jsou zaregistrováni na stejném serveru. Jak jsme si popisovali výše, volaný může být zaregistrován i u jiného serveru. Pak by zobrazený proxy server musel směřovat žádost INVITE na jiný proxy server, který by měl volaného na starost.



Obr.3: Průběh spojení podle SIP

V architektuře SIP není pro navázání komunikace server bezprostředně nutný, koncová zařízení se mohou kontaktovat přímo sami mezi sebou. Takovéto řešení není příliš vhodné, zejména kvůli adresování, kdy musí mít všichni klienti pevnou IP adresu nebo být rozlišováni pomocí DNS.

3 SOFTWAREVÁ ÚSTŘEDNA ASTERISK

3.1 Popis PBX Asterisk

Asterisk je kompletní softwarová pobočková ústředna (PBX) s otevřeným zdrojovým kódem pod GNU licenci. Je tvořena programovým vybavením poskytujícím služby všem standardním telefonním zařízení. Na rozdíl od běžné pobočkové ústředny jsou možnosti využití Asterisku daleko širší. Umožňuje např. propojit jednotlivé lokální telefonní sítě pomocí Internetu, připojit zaměstnance pracujícího doma do firemní telefonní sítě, poskytnou uživatelům funkci telefonní schránky nebo interaktivních hlasových aplikací.

Softwarová ústředna Asterisk byla primárně vyvinuta pro operační prostředí Linux, ale lze ji provozovat i pod operačními systémy Mac OS X, OpenBSD, FreeBSD nebo SunSolaris. Aktuální a mnou stažená verze Asterisku v době vzniku této práce byla 1.4.19. Na rozdíl od verze 1.2 přináší mnoho novinek a vylepšení. Vývojová větev verze 1.2 však stále není uzavřena. Existuje už i beta verze pro operační systémy Microsoft Windows.

Tato ústředna má obrovskou podporu spousty protokolů, mezi něž patří např. SIP, H.323, IAX, Cisco Skinny atd. Pro sadu protokolů H.323 poskytuje Asterisk služby koncového bodu (terminálu) a brány (gateway). U protokolu SIP zajišťuje ústředna Asterisk proxy server, redirect server a registrar server. Nesporně velmi cennou výhodou této ústředny je vlastnost fungovat jako brána mezi těmito protokoly.

Ať už možnost poskytování podrobné správy volání, vyúčtování jednotlivých volání nebo nemalá podpora na poli hardwarových zařízení (např. karty pro analogovou telefonní síť, ISDN karty atd.) patří mezi velkou škálu pozitivních vlastností, kterými tato PBX disponuje [4].

Asterisk podporuje následující kodeky:

- ADPCM - přenáší rozdíly mezi jednotlivými vzorky
- iLBC - Internet Low Bitrate Codec
- LPC10 - lineárních predikce
- GSM - asi nejlepší komprese pro přenos hlasu
- G.711 μ -law - kodek používaný pro Japonsko a Severní Ameriku
- G.711 a-law - kodek používaný pro Evropu a zbytek světa
- G.723.1
- G.726
- G.729
- Speex

Asterisk může zastávat funkci transkodéru mezi jednotlivými těmito kodeky, což ale klade nároky na výpočetní výkon.

3.2 Nároky na hardware

Nejprve je důležité si uvědomit, že nároky na hardware pro tuto ústřednu se budou lišit podle toho, co od ústředny očekáváme za práci. Např. pokud nám bude Asterisk pouze spojovat jednotlivé hovory pomocí protokolu SIP, není třeba velký výpočetní výkon. Jestliže však budeme po ústředně vyžadovat náročnější operace a to ať už typu konverze mezi různými kodeky, poskytování hlasové schránky, přehrávání hudby při čekání na přijetí hovoru (funkce Music On Hold) nebo aby vykonávala funkci hlasového či tónového automatu, logicky budeme muset na výpočetním výkonu přidat a nasadit ústřednu do mnohem výkonnějšího počítače. Po shrnutí předešlých kritérií se nedoporučuje instalovat Asterisk na stroje, které mají taktovací kmitočet menší než 500MHz. Počítač s touto frekvencí se dá považovat za mezní variantu a lze jej využít pro cca 10 klientů s využitím všech služeb ústředny. V praxi je ale lepší využívat počítače s 2GHz procesorem, pokud chceme využívat všechny funkce a vyhnout se možným komplikacím. Velkou výhodou je, že Asterisk svojí flexibilitou umožňuje spoustu pro nás nepotřebných součástí vypustit a upravit ústřednu tak, aby byla přesně pro naše potřeby a nároky [4].

3.3 Instalace potřebných částí

3.3.1 Linux

Jak už bylo řečeno ústředna Asterisk je primárně určena pro Linux, proto byl i mnou tento operační systém preferován. Nainstaloval jsem si distribuci Linuxu zvanou Ubuntu verzi 7.10. Každý uživatel by si měl především uvědomit, že Linux má jiný systém souborů než klasické Windows. Pro instalaci můžeme využít i náš stávající disk, na kterém už máme nainstalovaný jiný operační systém. Je jen nutné rezervovat část volného místa na disku pro Linux (minimálně kolem 5GB). Při takto paralelně nainstalovaných operačních systémech je třeba brát v potaz, že pokud chceme mít možnost nabootovat do Linuxu při spuštění systému, budeme muset nainstalovat klienta, který umožní vybrat při spuštění počítače operační systém, který chceme nabootovat. Linux potřebuje mít zavaděč na disku, který zavede jeho jádro do paměti.

Instalační průvodce nás postupně provede samotnou instalací, stačí se držet jeho kroků. Pro zkušenějšího uživatele je instalace relativně jednoduchá. Jakmile přijde na řadu rozdělení disku, doporučuji zvolit výše zmíněných 5GB pro kořenový adresář. Za další je zapotřebí vytvořit vlastní swap oddíl, ten by měl mít alespoň velikost rovnající se dvojnásobku velikosti naší RAM. Posledním důležitějším krokem je nastavit pojmenování počítače na název podle našeho jména v síti. Instalování dalších částí nebo rozšíření není nutné, zcela nám postačí základní systém s konzolí (terminálem). Instalaci potřebných balíčků a knihoven potřebných pro správný chod ústředny si popíšeme v následující části.

3.3.2 Používané balíčky

Předtím než započneme se samotnou instalací Asterisku, je třeba do systému zavést některé moduly a knihovny. Tyto moduly a knihovny je možné nainstalovat pomocí tzv. balíčků. Mezi hlavními moduly, které Asterisk využívá, jsou Zapata telephony (ovladače Zaptel) a PRI (knihovny libpri). Ovladače Zaptel slouží pro připojení analogového nebo digitálního hardware (např. karta TDM400P). Knihovna libpri slouží pro ISDN PRI rozhraní.

Asterisk je volně ke stažení na stránkách výrobce ve formátu `asterisk_verze.tar.gz` (soubor s příponou `*.tar.gz` je archiv původního zdrojového kódu programu, jak je distribuován vývojáři softwaru). Tento soubor je nutné zkompileovat GCC kompilátorem verze 3.00 nebo vyšší. Asterisk také potřebuje syntaktický analyzátor bison (nahrazuje yacc) a balíček ncurses pro funkčnost konzolového módu CLI. Modul Zaptel bude nutné nainstalovat, pokud budeme potřebovat využívat Ztdummy pro časování. Zaptel vyžaduje balíček libnewt [1 a 2].

Mnou nainstalované balíčky:

- bison
- g++
- libnewt-dev
- libncurses5-dev

3.3.3 Zaptel

Instalaci ovladačů Zaptel si lze stáhnout z oficiálního webu obdobně jako Asterisk www.asterisk.org. Po stažení je ho nutné nejprve rozbalit v adresáři `/usr/src` před samotnou instalací. Modul Zaptel musí být nainstalován dříve než se pustíme do instalace Asterisku, kdyby se instaloval až poté, ústředna by jej nenačetla. Zaptel budeme potřebovat jen ke zpřístupnění Ztdummy [1 a 2].

3.3.4 Ztdummy

Modul Ztdummy je virtuální zařízení navržené k tomu, aby poskytovalo zdroj hodinových pulzů v systému, kde není žádný hardware, který by to umožňoval. Ztdummy tedy zprostředkuje Asterisku časování. Ztdummy i Zaptel je po jejich instalaci nutné pro správnou funkci načíst [1 a 2].

3.3.5 Asterisk

Instalace ústředny Asterisk je opět volně ke stažení na oficiálním webu, kde je možno stáhnout si jak vývojové tak poslední stabilní verze. Já jsem využil nejnovější stabilní verzi 1.4.19. Jak už bylo uvedeno výše, po stažení je nutno Asterisk extrahovat. Pro umístění zdrojových souborů jsem zvolil adresář `/usr/src/`.

3.3.6 Použité příkazy

Jednotlivé výše uvedené části instalujeme pomocí následujících příkazů zadávaných v terminálu (pozn. je nutné pracovat pod uživatelem root pomocí příkazu `sudo` a mít tím pádem práva superuživatele. V operačním systému Linux nejdou jinak bez toho statusu instalovat veškeré programy a balíčky s knihovnamí) [1 a 2].

<code>#cd /usr/src/</code>	přepnutí do adresáře <code>/src</code>
<code># apt-get install <název_balíčku></code>	instalování příslušných balíčků
<code># wget <název_souboru></code>	stáhne z internetu požadovaný soubor
<code># tar zxgf <název_souboru></code>	provede dekomprimaci souboru
<code>#cd /usr/src/zaptel_verze</code>	instalace modulu Zaptel
<code>#make clean</code>	
<code>#./configure</code>	
<code>#make menuselect</code>	
<code>#make</code>	
<code>#make install</code>	
<code>#modprobe zaptel</code>	načtení modulu Zaptel
<code>#lsmod grep zaptel</code>	ověření načtení modulu Zaptel
<code>#modprobe ztdummy</code>	načtení modulu Ztdummy
<code>#lsmod grep ztdummy</code>	ověření načtení modulu Ztdummy
<code>#cd /usr/src/asterisk_verze</code>	instalace Asterisk
<code>#make clean</code>	
<code>#./configure</code>	
<code>#make menuselect</code>	
<code>#make</code>	
<code>#make install</code>	
<code>#make samples</code>	vytvoření vzorových konfiguračních souborů

3.4 Call centrum

Call centrum je organizační jednotka, která slouží k hromadnému zpracování příchozích a odchozích telefonních hovorů. Díky této službě, zprostředkované telefonní ústřednou, zákazníci ocení hlavně snadno dostupné informace, úsporu času nebo rychlou reakci a okamžité řešení problémů za pomoci vyškolených operátorů. Provozovatelé uvítají přínos této služby zejména v efektivnějším zpracování příchozích hovorů a optimálnějším rozložením pracovních sil v oddělení pracovníků, kteří zabezpečují fungování služeb a jiné technické záležitosti, od zákazníků.

Call centra musí disponovat dostatečným počtem telefonních linek (jejich počet se volí v zásadě podle vytížení ústředny). Mohou sloužit jak pro pasivní tak pro aktivní telemarketing.

Pasivní telemarketing - jde v podstatě o klasickou situaci, kdy operátoři přijímají příchozí hovory zákazníků, kteří volají například na infolinky. Tento druh péče o zákazníky je stále aktuální zejména v oblastech, kde je kontakt s "živým člověkem" více preferován např. linky technické podpory atd.

Využití: objednávkové linky, soutěžní linky, infolinky

Aktivní telemarketing - jedná se o účinnou formu oslovení, kdy operátoři kontaktují předem vybranou skupinu zákazníků za účelem například průzkumu trhu, pořádání telefonických anket či dotazníků, domlouvání schůzek nebo prodeji nových výrobků či služeb. Zpravidla se jedná o zpracování odchozích telefonních hovorů.

Využití: telefonické ankety, průzkumu trhu, aktuální nabídka služeb či produktů

Lze použít i kombinaci aktivního a pasivního telemarketingu, kdy přijatý hovor (pasivní telemarketing) je následně zpracován.

3.4.1 Automatické rozdělování hovorů

Technologie automatického rozdělování hovorů (ACD) určuje, jak se příchozí hovory rozdělují mezi jednotlivé pracovníky pracující v call centru. Například v malém call centru s třemi pracovníky péče o zákazníky jsme schopni pomocí ACD nadefinovat následující distribuci hovorů, kdy první příchozí hovor je směrován na linku jedna a mezi tím přichází další hovor, který je směrován na linku dva. Potom co je první hovor je ukončen, druhý stále přetrvává, přichází nový hovor, který je směrován na linku tři. ACD je tedy schopno například směrovat jednotlivé příchozí hovory na linku, která zůstala nejdelší dobu nečinná, čímž zajistí rovnoměrné rozložení hovorů mezi pracovníky call centra.

3.4.2 Volací fronty

Fronta nám dává logický prostor, kam odkládáme volající do doby, než jsme připraveni odpovědět na volání. Fronty jsou velmi flexibilní a silný nástroj k tomu, aby zlepšily zákaznický servis a lépe využily obsluhující personál. Můžeme u nich nastavovat spousty parametrů, mezi které patří například, že fronta může znít jako vyzvánění telefonu či používat hudbu na pozadí jako kulisu nebo může zákazníka čekající ve frontě informovat o jeho pozici či odhadovaném času čekání na spojení.

Volací fronty pracují s dvěma typy uživatelů. Prvním typem uživatelů jsou volající, kteří čekají ve frontě. Tito lidé mohou být chápáni jako zákazníci. Druhým typem uživatelů jsou členové, kteří pracují pro frontu. Tito lidé mohou být chápáni jako operátoři.

Pro zprovoznění volací fronty je důležité její nadefinování v konfiguračním souboru `queues.conf`. Podrobným nastavením tohoto souboru se budeme zabývat dále v textu. Na definovanou frontu je se třeba odvolat umístěním příslušné `extension` v souboru `extensions.conf`. Příklad `extensionu` volajícího frontu [1, 10]:

```
exten => 1000,1,Queue(nazev_fronty|parametr)
```

Při vytočení čísla 1000, bude volající umístěn do příslušné fronty. Položka parametru není nutnou součástí příkazu. Na její místo lze vložit spoustu doplňkových funkcí, které jdou libovolně kombinovat. Příklad těchto parametrů (jejich kompletnější rozpis [1]):

`n` – v případě nezastihnutí operátora je vykonána následující `extension` v pořadí
`r` – dává zákazníkovi vyzváněcí tón na místo automaticky hrající hudby na pozadí

Existují tři hlavní způsoby, jak lze připojit členy (operátory) do fronty a to staticky, dynamicky nebo používáním agentů. První možností je mít pro frontu členy staticky přidělené. V souboru `queues.conf` je pomocí příkazu `member` pevně nadefinován příslušný počet členů. Této varianty se nejvíce užívá u velkých front ústředěn. Druhou možností je přihlašování členů do fronty dynamicky pomocí aplikace `AddQueueMember()`. Příklad `extensionu` přihlašujícího členy:

```
exten => 1001,1, AddQueueMember(nazev_fronty)  
exten => 1002,1, RemoveQueueMember(nazev_fronty)
```

Při vytočení čísla 1001, bude přihlášen člen do příslušné fronty. Druhá `extension` volaná přes číslo 1002 slouží pro spuštění aplikace `RemoveQueueMember()`, která odhlásí člena z fronty. Aplikace `AddQueueMember()` však umí přeskočit na prioritu `n+101`, pokud je rozhraní již členem fronty. Této funkce se dá využít, tak že pro přihlášení i odhlášení lze použít vytočení pouze jednoho čísla:

```
exten => 1001,1,Answer
exten => 1001,2,AddQueueMember(nazev_fronty)
exten => 1001,3,Playback(agent-loginok)
exten => 1001,4,Hangup
exten => 1001,103,RemoveQueueMember(nazev_fronty)
exten => 1001,102,Playback(agent-loggedoff)
exten => 1001,105,Hangup
```

Třetí způsob jak lze přihlásit člena do fronty je pomocí takzvaného agenta. Jednotlivé profily agentů se nastavují v konfiguračním souboru `agents.conf`, kde je jim mimo jiné každému přiděleno identifikační číslo a heslo. Agenti popřípadě skupiny agentů jsou potom zapsáni pomocí příkazu `member` v konfiguračním souboru `queues.conf`.

Asterisk umožňuje práci dvěma druhům agentů. Jednak jsou to agenti, kteří pracují především ve velkých call centrech a jsou k dispozici po celou dobu, co jsou přihlášení k systému. Tento způsob je efektivnější, odpadá starost se zvedáním či přílišným vyzváněním telefonů. Agent, který je volný a čeká ve frontě na volání slyší hudbu na pozadí, do doby než přijme hovor. Příchozí hovor je oznámen agentovi příslušným zvukovým signálem a následně je s agentem ihned spojen. Agenti mohou být současně ve více frontách, před příchozím hovorem jim pak bývá ještě oznámeno, z které fronty hovor přichází.

```
exten => 1003,1,AgentLogin
```

Jedná-li se o call centrum s nižším provozem, kde zaměstnanci nesedí trvale u telefonního přístroje, je nežádoucí, aby příchozí hovor byl ihned spojen. V tomto případě se pro přihlášení agenta používá následující extension, při kterém příchozí hovor začne agentovi nejprve vyzvánět a ten je potom musí manuálně zvednout, aby došlo ke spojení.

```
exten => 1003,1,AgentCallBackLogin
```

Při vytočení čísla 1003, začne probíhat přihlášení agenta do systému. Bude vyzván k tomu, aby zadal svoje identifikační číslo a heslo. V případě příkazu `AgentCallBackLogin` bude systém ještě vyžadovat, aby byla zadána příslušná extension (číslo pobočky na které budeme brát volání, je nutné k tomu, aby nás Asterisk zpětně dohledal).

3.4.3 Hudba na pozadí

Většina moderních pobočkových ústředen disponuje nějakým zdrojem hudby, který je přehráván, pokud například čekají volající ve frontě na spojení. Samozřejmě i systém Asterisk službu hudba na pozadí, anglicky music on hold, podporuje. Výchozím formátem u této služby jsou pro Asterisk verze 1.4 nekomprimované WAV soubory.

V dnešní době patří mezi nejznámější hudební formáty soubory MP3. Tento formát však není pro užití jako hudby na pozadí vhodnější. U MP3 hudebních souborů se využívá zejména vlastnost velké úspory místa, které dosahují díky značné kompresi. Jejich malá velikost je ale vykoupena potřebou soubor před samotným přehráním dekodovat, čemuž je zapotřebí značného výpočetního výkonu. U dnešních ústředen není ani tak problémem dostatek místa, jako právě zmíněná potřeba výpočetního výkonu. Pokud by nastala například situace, kdy je potřeba vysílat hudbu do mnoha kanálů (např. velké call centrum se spoustou čekajících lidí ve frontě nebo linky na kterých lidé poslouchají hudbu přes jejich telefon atd.), pak by dekodování MP3 souborů zabralo značnou část výpočetního výkonu, který by jinak mohl být systémem využit pro jiné operace. Proto je vhodné převést hudební zdroje na nativní formáty různých kodeků, které ústředna podporuje (např. pokud používáme a-law pro pevné linky a G.729 pro VoIP zařízení, je vhodné mít hudební soubory v obou dvou formátech pro oba tyto kanály, Asterisk nebude muset vykonávat převodní funkce).

Pokud se přece jen potřeba použít některý z komerčních kompresních formátů MP3 nebo OGG (má stejnou koncepci jako MP3, není s ní však kompatibilní), je vhodné je přeměnit na formát, který umí Asterisk snadno ovládat. K tomu slouží například utilita Sound eXchange, lze nainstalovat pomocí příkazu:

```
#apt-get install sox
```

Pokud je utilita nainstalována, jednotlivé MP3 nebo OGG soubory se převádí na formát WAV následujícím příkazem:

```
#sox nazev_souboru.ogg nazev_souboru.wav
```

U tradičních ústředen je hudba na pozadí pouštěna obvykle z jednoho zdroje, to znamená, že každý člověk slyší ve stejném čase to samé a dokonce když není nikdo komu jí pouštět, hudba běží stále dál. Ústředna Asterisk přehrává každému volajícímu vlastní hudební zdroj, který je aktivní jen po dobu, jakou je potřeba.

Asterisk umožňuje přehrávat i více souborů a to buď to v daném pořadí za sebou nebo zcela náhodně po sobě. Zvukové soubory, které chceme použít nahrajeme všechny do jednoho adresáře a odkážeme se na něj v konfiguračním souboru musiconhold.conf. Náhodný výběr skladeb se realizuje pomocí funkce *random*. Toto nastavení ocení hlavně ti, kteří často volají do call centra [1].

3.4.4 Hlasová samoobsluha

Kdykoliv voláme nějakou větší společnost, ve většině případů nás uvítá hlas záznamníku, který se nás snaží nasměrovat dle naší potřeby. Obeznámí nás s možnostmi volby (tzv. volací menu), v rámci něhož si potom volíme služby podle našeho přání. Tento systém volby se nazývá hlasová samoobsluha, anglicky Interactive Voice Response (IVR). Jejím hlavním úkolem je komfortně a co nejjednodušším způsobem nasměrovat uživatele na základě jeho požadavků. Následujícími příkazy si ukážeme, z čeho se jednoduchá samoobsluha skládá. Příkazy se zapisují do konfiguračního souboru extensions.conf [6 a 2].

```
exten => 100,1,Answer()
exten => 100,n,DigitTimeout,10
exten => 100,n,ResponseTimeout,30
exten => 100,n,Playback(${cesta_k_souboru}uvitani)
exten => 100,n,Wait(1)
exten => 100,n,Playback(${cesta_k_souboru}volba1)
exten => 100,n,Wait(1)
exten => 100,n,Playback(${cesta_k_souboru}volba2)
exten => 100,n,Wait(1)
exten => 100,n,Playback(${cesta_k_souboru}volba3)
exten => 1,1,Goto(volba1,s,1)
exten => 2,1,Goto(volba2,s,1)
exten => 3,1,Goto(volba3,s,1)
exten => t,1,Hangup()
```

```
[volba1]
exten => s,1,Dial(SIP/volba1)
```

```
[volba2]
exten => s,1,Dial(SIP/volba2)
```

```
[volba3]
exten => s,1,Dial(SIP/volba3)
```

Vytvořil jsem modelovou situaci nastavení hlasové samoobsluhy. Při vytočení volacího čísla 100 uživatele přivítá uvítací zprava. Poté jsou mu pomocí hlasového menu nabídnuty jednotlivé volby (např. při stisknutí 1 přesměrování na volbu1 atd.) Uživatel má 30 sekund na svou volbu, pokud tato doba vyprší, hovor je automaticky ukončen.

3.5 Konfigurace PxB Asterisk

Ústřednu Asterisk konfigurujeme pomocí textových konfiguračních souborů, které se nacházejí v adresáři `/etc/asterisk/` (pokud jsme při instalaci použili příkaz *make samples*). V tomto adresáři nalezneme všechny důležité konfigurační soubory:

- `agents.conf` - konfigurace agentů
- `sip.conf` - konfigurační soubor SIP
- `queues.conf` - konfigurace volacích front
- `musiconhold.conf` - konfigurace hudby na pozadí
- `extensions.conf` - konfigurace Dialplanu

Za další důležité soubory ke konfiguraci lze považovat `zapata.conf` a `zaptel.conf`. Tato dvojice konfiguračních souborů slouží pro nastavení Asterisku, pokud se ho chystáme připojit k PSTN [9]. Jejich podrobným nastavením se nebudeme zabývat, protože pro nás nemají smysl.

3.5.1 Agents.conf

V tomto souboru se vytvářejí a konfigurují jednotliví agenti pro call centrum a aplikace fronty. Parametry v sekci `[agents]` definují jednotlivé agenty a určují jakým způsobem s nimi systém bude nakládat. Nastavení se vztahují ke všem agentům, pokud nemá agent u sebe definováno jinak [10].

<i>ackcall</i>	může nabývat hodnot <i>yes</i> nebo <i>no</i> , udává zda je třeba pro přijetí hovoru stisknout tlačítko # či nikoliv
<i>autologoff</i>	udává se v sekundách a určuje po jaké době nečinnosti má být agent automaticky odpojen
<i>wrapuptime</i>	udává se v milisekundách a určuje po jaké době se agent po ukončení hovoru znovu k dispozici
<i>musiconhold</i>	nastavuje hudbu na pozadí

Agenti se konfigurují následovně, položka *group* slučuje více agentů do skupin je-li třeba (není povinná):

```
group = cislo_skupiny
agent => agentovo_ID, agentovo_heslo, agento_jmeno
```

3.5.2 Sip.conf

Tento soubor obsahuje konfiguraci všech zařízení komunikujících s ústřednou Asterisk pomocí SIP protokolu. Tato komunikace probíhá ve většině případů hlavně mezi telefony a to ať už hardwarovými nebo softwarovými. Obsah tohoto souboru se skládá z dvou hlavních částí. První částí je sekce [general], kde se nacházejí všeobecné nastavení platné pro všechny telefony (např. nastavení portu, kodeku...). Součástí konfigurace může být i sekce [globals], která slouží pro definici proměnných. Druhou částí jsou už jednotlivé, přesně specifikované profily konkrétních uživatelů. Důležitým parametrem je tzv. *context*, na který se odvoláváme v *extensions.conf*, kde určuje jaký způsob přijetí hovoru uživateli přiřadí. [9]

<i>port</i>	určuje port pro komunikaci (5060 SIP standardně používá)
<i>disallow</i>	zakazuje určité kodeky, při přiřazení parametru <i>all</i> zakáže všechny
<i>allow</i>	povoluje určité kodeky (funguje jen v kombinaci s <i>disallow</i>)
<i>context</i>	nastavuje kontext
<i>username</i>	nastavuje uživatelské jméno
<i>secret</i>	nastavuje uživatelské heslo
<i>host</i>	nastavuje IP adresu telefonu (není-li pevná IP, píšeme <i>dynamic</i>)
<i>type</i>	může nabývat hodnot <i>user</i> , <i>peer</i> a <i>friend</i> , udává zda komunikace může probíhat v obou směrech či jen v jednom

3.5.3 Musiconhold.conf

Konfigurační soubor v kterém nastavujeme hudbu na pozadí a její parametry. Názvy jednotlivých profilů píšeme do hranatých závorek [11].

<i>mode</i>	určuje hlasitost hudby, jestli má být ve vyrovnávací paměti atd.
<i>directory</i>	nastavuje adresář obsahující zdrojové hudební soubory
<i>random</i>	může nabývat hodnot <i>yes</i> nebo <i>no</i> , udává zda je přehrávání hudebních souborů náhodné či ne

3.5.4 Queues.conf

Tímto souborem nastavujeme a vytváříme jednotlivé volací fronty. Skládá z následujících částí, sekce [generals] definující nastavení, která se aplikují pro všechny fronty. Další sekce jsou již přímo jednotlivé fronty. Názvy jednotlivých front píšeme do hranatých závorek [1, 11].

<i>persistentmembers</i>	může nabývat hodnot <i>yes</i> a <i>no</i> , ukládá všechny dynamicky přidané členy ve všech frontách, pokud zvolíme parametr <i>yes</i> a dojde k restartu ústředny členové budou automaticky znovu načtení (defaultně nastaveno)
<i>maxlen</i>	udává maximální počet volání ve frontě (0 znamená bez limitu)
<i>emptyjoin</i>	může nabývat hodnot <i>yes</i> , <i>no</i> a <i>strict</i> , udává jestli se mohou volající připojit do fronty v případě nepřítomnosti agentů
<i>timeout</i>	udává se v sekundách a určuje po jaké době, bude agent považován za nedosažitelného a volající se zařadí zpět do fronty
<i>retry</i>	udává se v sekundách a určuje po jaké době v případě překročení hodnoty <i>timeout</i> bude agentovi opět směřován další volající
<i>wrapuptime</i>	udává se v sekundách a určuje po jaké době po úspěšném hovoru je agentovi opět směřován další hovor
<i>announce-frequency</i>	udává se v sekundách a určuje po jaké době má být volající informován o místě ve frontě a odhalovaném času do spojení (0 znamená deaktivaci)
<i>announce-holdtime</i>	může nabývat hodnot <i>yes</i> , <i>no</i> a <i>once</i> , udává jestli se má oznamovat odhadovaný čas do spojení v oznamování
<i>musiconhold</i>	nastavuje hudbu ve volací frontě (pokročilejší nastavení v souboru <i>musiconhold.conf</i>)

Asterisk umožňuje šest různých možností distribucí volání (ACD), které udávají jakou formou bude probíhat distribuce hovoru. Nastavuje se příkazem *strategy* a patřičným parametrem:

<i>ringall</i>	fronta volá každého dostupného agenta, spojuje volání s tím co první odpoví (defaultně nastaven)
<i>roundrobin</i>	fronta cyklicky pojíždí mezi agenty, do té doby než nalézá volného agenta, nebere v úvahu vytíženost agentů a začíná vždy od prvního
<i>leastrecent</i>	fronta volá agenta, který byl v poslední době nejméně ze všech vytížený
<i>fewestcalls</i>	fronta volá agenta, který byl v poslední době nejméně ze všech volaný
<i>rrmemory</i>	fronta se chová stejně jako <i>roundrobin</i> jen si navíc pamatuje koho naposled volala a začíná od něj místo, aby začala od prvního
<i>random</i>	vybírání agenta z fronty zcela náhodně

Přiřazování členů do fronty se provádí pomocí příkazu *member*, který se zapisuje následovně. Číslo penalizace není nezbytnou součástí příkazu. Udává jak často je člen volán, přičemž největší číslo znamená, že je člen volán naposledy:

`member => SIP/3001,cislo_penalizace`

`member => Agent/agentovo_ID`

Asterisk může také připojovat celé skupiny agentů. Toto je výhoda zejména u velkých volacích front, kde se tímto vyřeší dlouhé vypisování stovek agentů. Lze zapsat dvěma způsoby:

`member => Agent/@1`

`member => Agent/:1`

Uvedené příkazy přiřazují všechny agenty první skupiny jako členy do fronty.

3.5.5 Extensions.conf

Tento soubor lze považovat za jádro celého systému, v jeho režii se odehrává veškeré chování všech spojení. Ať už se jedná o libovolné uskutečněné spojení prostřednictvím ústředny, právě v tomto souboru Asterisk přiřadí tomuto spojení patřičný Dialplan (souhrn po sobě jdoucích příkazů, které mají na starost řízení spojení). Tak jako předešlý konfigurační soubor i tento začíná sekcí [general] pro všeobecné konfigurační informace a opět sekcí [globals] pro definici proměnných. Dále už soubor pokračuje jednotlivými uživatelskými kontexty. Uživatel začíná na kontextu, který má definovaný ve svém profilu v *sip.conf* (pokud není nadefinován, začíná na kontextu nastaveném v sekci [general]). V konfiguraci u toho souboru *extensions.conf* se setkáme převážně s dvojicí příkazů a to *include* a *exten*. Příkazem *include* používáme ke vkládání do daného kontextu obsah jiného kontextu. Příkaz *exten* používáme k vytvoření Dialplanu. Tyto příkazy mají následující syntaxi [9]:

`include => název_kontextu`

`exten => číslo,priorita,příkaz (parametry)`

Příkaz *exten* začíná většinou volaným číslem, popřípadě speciálním jednopísmenovým znakem (např. s,i,h...). Priorita určuje pořadí jednotlivých *exten* příkazů jak jdou po sobě. Stačí napsat první číslo a pak psát n (automaticky doplňuje další čísla s přírůstkem jedna). Příkaz je pak už konkrétní požadavek co se má provést.

Speciální jednopísmenové znaky:

s – daný příkaz se provede automaticky, hned při vstupu do daného kontextu

i - daný příkaz se provede v případě špatné volby

h - daný příkaz se provede v případě zavěšení hovoru

t - daný příkaz se provede v případě překročení časového limitu

Při konfigurování souboru *extensions.conf* jsem do sekce [general] připsal dvojici příkazů *static* a *writeprotect* jejichž parametry jsem nastavil na hodnotu *yes*. Tyto dva příkazy zabraňují možnosti dynamické změny Dialplanu.

3.6 Spouštění Asterisku

Po instalaci a konfiguraci ústředny nám již nic nebrání v jejím samotném spuštění. Asterisk spouštíme z adresáře `/etc/asterisk/` pomocí příkazu `asterisk` a patřičného parametru. Parametrem specifikuje jak se má ústředna spustit. Příkazem `asterisk -h` obdržíme kompletní výpis všech dostupných parametrů. Mezi nejčastěji používané patří [1]:

- V vypíše verzi Asterisku kterou používáme
- c spouští Asterisk v tzv. konzolovém módu CLI
- g výpisy jádra v případě kolize
- r spuštění Asterisku v normálním módu
- T zobrazí čas ve formátu [Mmm dd hh:mm:ss] pro každý řádek v konzolovém módu CLI
- v zvyšuje počet výpisů jednotlivých událostí ústředny (čím více „v“ napíšeme tím podrobnější)
- x <příkaz> provede daný CLI příkaz (funguje pouze s parametrem -r)

3.6.1 Konzolový mód CLI

V konzolovém módu Asterisku můžeme zadávat různé příkazy pro efektivní spravování ústředny. Tento mód využíváme zejména při ladění ústředny, již zaběhnutý systém aktivujeme spíše pomocí parametru `-r`. Jednotlivé příkazy zadáváme do příkazového řádku `*CLI>` Mnou nejpoužívanější příkazy byly:

- `help` vypíše všechny příkazy
- `help <příkaz>` vypíše návod k použití příkazu
- `restart now` restartuje ústřednu (znovu načte konfiguraci)
- `stop now` vypne ústřednu
- `agent show` vypíše agenty a jejich stav
- `core show channeltypes` vypíše seznam dostupných kanálů
- `moh show classes` vypíše všechny třídy hudby na pozadí
- `moh show files` vypíše hudební soubory hudby na pozadí
- `queue show` vypíše fronty a jejich stav
- `sip show users` vypíše všechny SIP klienty

4 SOFTWAREVÝ KLIENT X-LITE

4.1 Popis X-Lite

X-Lite od společnosti CounterPath je softwarový klient, nabízející uživateli kompletní komfort klasického telefonu s využitím počítačového rozhraní. Tento softwarový telefon využívá ke komunikaci protokol SIP. X-Lite poskytuje koncový bod VoIP řešení, které může využívat např. Internet, servery telefonie uvnitř podnikové LAN nebo třeba VoIP poskytovatele. K výhodám programu X-Lite nesporně spadá také velká podpora kodeků (např. G.711, iLBC, GSM atd.) a v neposlední řadě i spousta zajímavých funkcí jako jsou například konferenční hovory, video hovory či nahrávání hovorů a jiné.

Tento softwarový telefon běží jak pod operačním systémem Windows, tak ho lze uplatnit s patřičnou instalací i u systému MAC OS X. Minimální požadavky na počítač, aby program zaručoval bezstarostný chod jsou procesor s taktovací frekvencí minimálně 700MHz, velikost RAM 256MB a volné místo 30MB na pevném disku. Doporučené požadavky na procesor jsou však 2GHz. Zvuková karta a připojení na Internet jsou samozřejmostí. Více o tomto programu se lze dozvědět u výrobce [10].

Mnou používaná verze programu byla 3.0 build 41150

4.2 Základní nastavení

Po spuštění se objeví základní menu programu (viz Obr.4). Kliknutím na tlačítko menu se objeví kontextové menu, v kterém si vyberte možnost *SIP Account settings* (nastavování SIP klientů). Zde provedeme základní nastavení programu, tím že vyplníme následující povinné položky:

- Display name - vaše jméno
- User name - přidělené číslo (nakonfigurováno v sip.conf Asterisku)
- Authorization user - přidělené číslo (nakonfigurováno v sip.conf Asterisku)
- Password - přidělené heslo (nakonfigurováno v sip.conf Asterisku)
- Domain - IP adresa serveru

Další nastavování programu jsem neprováděl, program plně fungoval již v základním nastavení po instalaci.

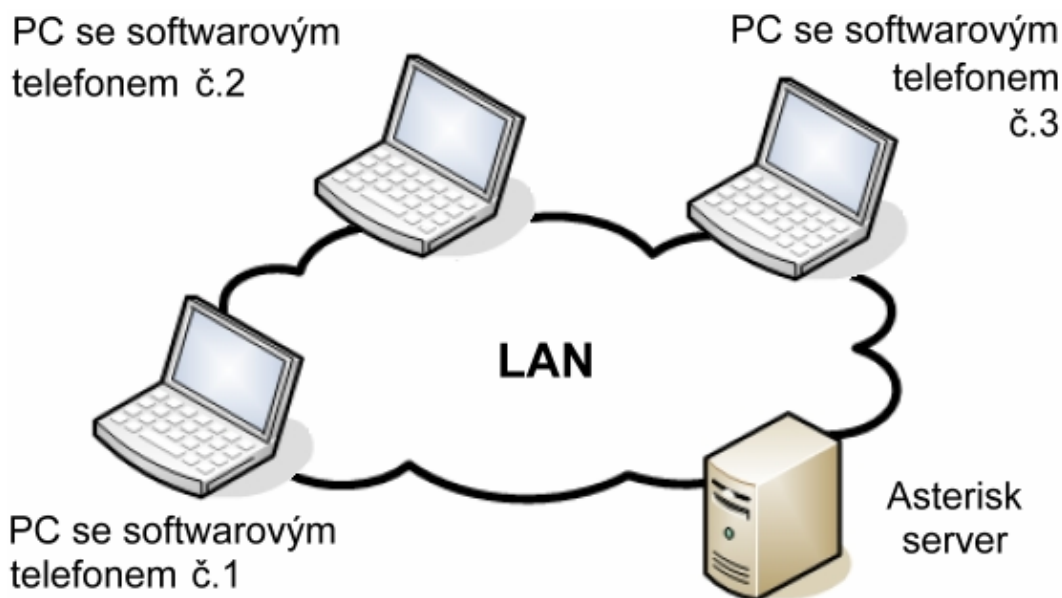


Obr.4: Popis základních funkcí X-Lite

5 PRAKTICKÉ ODZKOUŠENÍ

5.1 Testovací zapojení

Do běžného osobního počítače, bez přidaného jakéhokoliv dalšího speciálního hardware, jsem na instaloval operační systém Linux distribuci Ubuntu verzi 7.10. Na linuxovém uživatelském rozhraní jsem následovně nainstaloval pobočkovou ústřednu Asterisk verze 1.4.19 ještě spolu s dalšími komponentami potřebnými k její bezchybné funkci. Samotné spuštění Asterisk proběhlo bez komplikací. Ústřednu jsem nainstaloval na jeden ze čtyř počítačů uvnitř sítě LAN. K vlastnímu otestování ústředny jsem tedy použil zbývající tři počítače v síti. Na každý ze zbývajících počítačů jsem nainstaloval a spustil softwarový telefon X-Lite verzi 3.0 a zaregistroval se k ústředně na patřičné mnou nastavené SIP účty. Úspěšně bylo provedeno několik zkušebních hovorů mezi stanicemi vedených přes ústřednu. Testovací zapojení je znázorněno na Obr.5.



Obr.5: Testovací zapojení

Následně jsem se telefonem č.1 přihlásil jako agent a byl přiřazen jako člen do mnou vytvořené volací fronty, zbylými dvěma klienty sem tuto frontu volal. Při dovolání do fronty začala na telefonech č.2,3 hrát zvolená hudba na pozadí. Pro hudbu jsem nastavil zdrojový adresář, který obsahoval kolem dvaceti skladeb. Nastaven byl náhodný výběr skladeb, což zajišťovalo, že každé volání fronty uvítala skoro pokaždé jiná skladba. Jakmile telefony č.2,3 vstoupily do fronty, telefon č.1, zaregistrovaný v systému jako agent, začal vyzvánět. Při zvednutí hovoru bylo uskutečněno spojení mezi telefony. První byl s agentem spojen telefon č.2, protože se do fronty zařadil dříve než telefon č.3. Po ukončení hovoru přišel na řadu telefon č.3. Agentovi začal telefon vyzvánět až po nastavené prodlevě, která určovala dobu, kdy je agent po úspěšném spojení opět přístupný dalším voláním zařazeným do fronty.

Prověřil jsem i ostatní parametry nastavení fronty, kdy jsem agenta nechal vyzvánět. Po nastavené časové prodlevě byl agent brán za nedostupného a volání bylo zařazeno zpět do fronty. Následovala doba, která určovala, kdy má být agent opět volán pokud byl jeho předešlý hovor neúspěšný.

Posledním testovaným parametrem fronty bylo nastavení její distribuce hovorů, čili automatické třídění hovorů mezi více agentů. Moje fronta volala agenta, který byl v poslední době nejméně ze všech vytížený. K úplnému prověření korektnosti této distribuce hovorů by bylo potřeba minimálně třech agentů, bohužel z důvodů omezeného množství telefonů v síti, nebylo možné toto dostatečně vyzkoušet. Telefony č.1,2 jsem přihlásil jako agenty fronty. Telefonem č.3 jsem následovně frontu volal. Nejprve vyzváněl agent na telefonu č.1. Po ukončení spojení jsem telefonem č.3 opět volal frontu, hovor byl automaticky přidělen agentovi, jenž byl v poslední době nejmíň nečinný, tudíž agentovi na telefonu č.2. Všechna volání byla uskutečněna bez jakýkoliv komplikací.

5.2 Konfigurace ústředny

Ústředna se konfiguruje pomocí souborů umístěných v adresáři `/etc/asterisk/` viz. uvedeno výše. Nastavoval jsem konfigurační soubory `agents.conf`, `sip.conf`, `queues.conf`, `musiconhold.conf` a `extencions.conf`. Podrobné nastavení jednotlivých souborů je uvedeno na konci této práce v příloze 2 a v elektronické podobě vypálené na přiloženém CD.

5.3 Použité technické vybavení

Hardware

osobní počítač

procesor:	Intel Core2Duo 2,26GHz
operační paměť:	1024MB

Software

operační systém:	Linux distribuce Ubuntu verze 7.10
softwarová PBX:	Asterisk verze 1.4.19
softwarový telefon:	X-Lite verze 3.0 build 41150

ZÁVĚR

Hlavním cílem této bakalářské práce bylo seznámit se s problematikou IP telefonie a softwarovou telefonní ústřednou Asterisk podporující její služby. Práce měla za úkol ověřit možnost nasazení této ústředny do provozu call centra, a to hlavně se zaměřením na vytvoření a správu volacích front. Dosažené výsledky této práce jsou prezentovány v poslední kapitole, kde je praktické odzkoušení ústředny podrobněji popsáno. Ústřednu se podařilo nakonfigurovat, tak aby splňovala požadované body zadání.

Dle mého názoru po zjištění všech výhod, kterými Asterisk disponuje, si myslím, že se jedná o perspektivní systém jehož uplatnění (ať již u malých nebo velkých firem) je obrovské. Hlavními klady jsou bezesporu maximální flexibilita a možnost nasazení prakticky do každého osobního počítače. Co se konfigurace a správy ústředny týče, jednotlivé bloky mají širokou škálu možností nastavení. Spousta funkcí je však jen doplňková, ty hlavní si lze osvojit za pár dní. Ústředna Asterisk na mě zapůsobila jako moderní a profesionální řešení vhodné všude tam, kde je zapotřebí služeb pobočkové ústředny.

LITERATURA

- [1] MEGGELEN, J.V, SMITH, J., MADSEN, L. *Asterisk™: The Future of Telephony*. 2nd Edition, printed in the United States of America 2005, ISBN 0-596-51048-9
- [2] GOMILLION, D., DEMPSTER, B. *Building Telephony Systems with Asterisk*. Packt Publishing, 2005, ISBN 978-1904811152
- [3] VRBA, K., NAGY, Z., *Multimediální služby*. Skripta FEKT VUT v Brně
- [4] CHOVANEC, T. *IP telefonie v prostředí embedded systémů*. [s.l.], 2007. 64 s. Diplomová práce.
- [5] BUČEK, O. *Internetová telefonie pro malou organizaci*. [s.l.], 2006. 59 s. Bakalářská práce.
- [6] WIJA, T., ZUKAL, D., VOZŇÁK, M. *Asterisk a jeho použití*. [s.l.], 2005. 38 s. Technická zpráva.

Odkazy na internetu:

- [7] *Wikipedia* [online]. 2001-2007 [cit. 2007-10-25]. Dostupný z WWW: <<http://wikipedia.org>>
- [8] HRUBÝ, P. *ISDN Server* [online]. 2003 [cit. 2007-10-30]. Dostupný z WWW: <<http://www.isdn.cz>>
- [9] VALOUŠEK, O. *ABC Linuxu* [online]. 2006 [cit. 2007-12-01]. Dostupný z WWW: <<http://www.abclinuxu.cz>>
- [10] *Asterisk Guru* [online]. 2008 [cit. 2008-03-18]. Dostupný z WWW: <<http://www.asteriskguru.com>>
- [11] *Voip-info.org* [online]. 2008 [cit. 2008-01-27]. Dostupný z WWW: <<http://www.voip-info.org>>
- [12] *X-Lite* [online]. 2007 [cit. 2007-11-21]. Dostupný z WWW: <<http://www.counterpath.com>>

SEZNAM PŘÍLOH

Příloha 1: Laboratorní úloha

Příloha 2: Ukázky konfiguračních souborů

PŘÍLOHA 1: LABORATORNÍ ÚLOHA

Konfigurace pobočkové ústředny Asterisk

Cíl měření:

Laboratorní úloha umožňuje studentům na základě konfigurace jednotlivých souborů Asteriku získat praktickou zkušenost a základní poznatky, jak nastavit tuto softwarovou pobočkovou ústřednu. Během nastavování se studenti seznámí nejen s konfiguračními soubory ústředny, ale i s klientem X-Lite a signalizačním protokolem SIP.

Zadání:

- 1) Seznamte se signalizačním protokolem SIP, softwarovou ústřednou Asterisk a principem volacích front.
- 2) Nastavte účty SIP klientů, vytvořte agenta a volací frontu.
- 3) Přihlaste se jako agent a vyzkoušejte volací frontu.
- 4) Zkoušejte měnit jednotlivé parametry v konfiguračních souborech a pozorujte jak se projeví změny

Teoretický úvod:

Signalizační protokol SIP

V dnešní době je více nasazován protokol SIP oproti staršímu H.323. Je používán jako signalizační protokol. K vlastnímu přenosu používá jiné existující standardy. Hojně je využit v IP telefonii, kde má na starost sestavení, modifikaci a ukončení spojení mezi dvěma a více uživateli. Spojení může ale obecně představovat i jakýkoliv jiný multimediální přenos, ne jen IP telefonii. SIP protokol je nedílnou součástí architektury TCP/IP. Za normálních podmínek používá UDP, ale funguje i na TCP. Protokol SIP je založen na textové bázi. Využívá principy známé z Internetových protokolů jako jsou HTTP a nebo poštovní protokol SMTP. Je velmi flexibilní, umožňuje jednoduché rozšiřování. Tento protokol pracuje na principu klient – server. Při této komunikaci dochází k výměně dvou typů zpráv (metoda – odpověď).

Pobočková ústředna Asterisk

Jedná se o kompletní softwarovou ústřednu s otevřeným zdrojovým kódem pod GNU licenci. Je tvořena programovým vybavením poskytujícím služby všem standardním telefonním zařízením. Je primárně vyvinuta pro operační prostředí Linux, ale lze ji provozovat i pod operačními systémy Mac OS X, OpenBSD, FreeBSD nebo SunSolaris. Tato ústředna má obrovskou podporu spousty protokolů, mezi něž patří např. SIP, H.323,

IAX atd. Pro sadu protokolů H.323 poskytuje Asterisk služby koncového bodu (terminálu) a brány (gateway). U protokolu SIP zajišťuje ústředna proxy server, redirect server a registrar server. Nesporně velmi cennou výhodou této ústředny je vlastnost fungovat jako brána mezi těmito protokoly.

Volací fronty

Fronta nám dává logický prostor, kam odkládáme volající do doby, než jsme připraveni odpovědět na volání. Fronty jsou velmi flexibilní a silný nástroj k tomu, aby zlepšily zákaznický servis a lépe využily obsluhující personál. Můžeme u nich nastavovat spousty parametrů, mezi které patří například, že fronta může znít jako vyzvánění telefonu či používat hudbu na pozadí jako kulisu nebo může zákazníka čekající ve frontě informovat o jeho pozici či odhadovaném času čekání na spojení.

Volací fronty pracují s dvěma typy uživatelů. Prvním typem uživatelů jsou volající, kteří čekají ve frontě. Tito lidé mohou být chápáni jako zákazníci. Druhým typem uživatelů jsou agenti, kteří pracují pro frontu. Tito lidé mohou být chápáni jako operátoři.

Postup měření:

- 1) Na počítači, kde je nainstalována ústředna Asterisk, editujte dle návodu postupně její konfigurační soubory sip.conf, agents.conf, queues.conf a extensions.conf, které se nachází v adresáři /etc/asterisk/

sip.conf.....tento soubor slouží pro nastavování SIP účtů

```
[general]                ;sekce definující všeobecná nastavení
port = 5060              ;port přes který SIP komunikuje
context = hlavni        ;nastavení kontextu

[100]                   ;učet klienta 100
type = friend           ;směru komunikace
username = 100          ;uživatelské jméno
secret = 1234           ;uživatelské heslo
host = dynamic          ;nastavení IP adresy
```

stejným způsobem vytvořte ještě jeden SIP účet pro klienta [200]

queues.conf.....tento soubor slouží pro nastavování volacích front

```
[fronta]                ;nastavení fronty fronta
timeout = 15            ;po překročení vrací volajícího zpět do fronty
retry = 5               ;za jak dlouho opět volat agenta při neúspěchu
wrapuptime = 0         ;za jak dlouho opět volat agenta při úspěchu

member => Agent/007    ;přiřazení agenta frontě
```

agents.conf.....tento soubor slouží pro nastavování agentů

```
[Agents] ;sekce definující agenty
ackcall = no ;zda je k přijetí hovoru třeba tlačítko #

agent => 007,1234,Agent ;přidání agenta (ID, heslo, jméno)
```

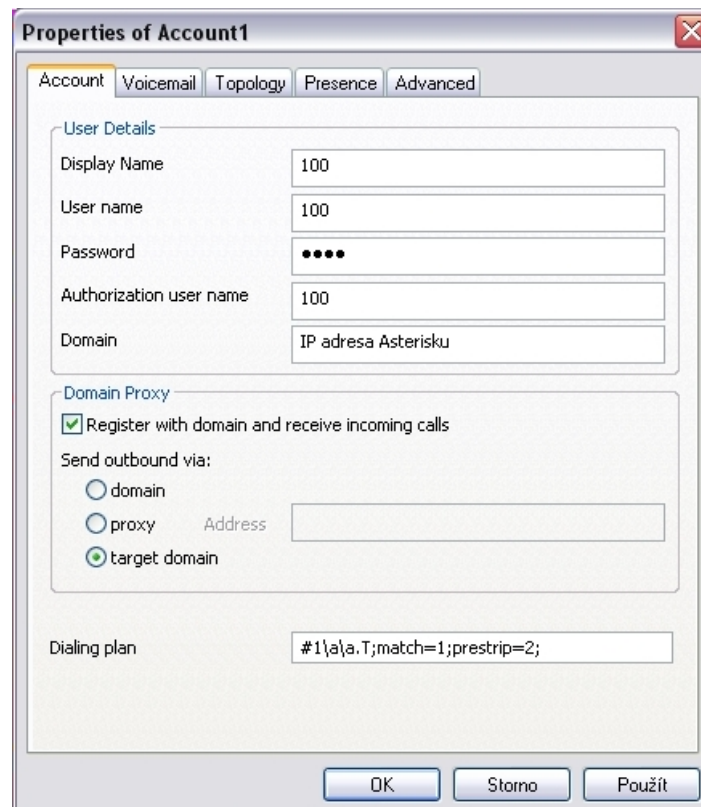
extensions.conf.....tento soubor slouží pro nastavování jednotlivých spojení

```
[hlavni]
exten => 112,1,Answer ;\
exten => 112,2,Ringing ; \
exten => 112,3,Wait(5) ; > Dialplan pro volací frontu
exten => 112,4,Queue(fronta) ; /
exten => 112,5,Hangup ;/

exten => 999,1,AgentCallbackLogin(||@agent) ;log in/out agentů

[agent]
exten => 100,1,Dial(SIP/100)
```

- 2) Po nastavení konfiguračních souborů ústřednu spustíme pomocí příkazu *asterisk -r*
- 3) Spustíme softwarový telefon X-Lite na dalších počítačích a připojíme na vytvořené SIP účty 100 a 200 (viz. Obr.1).



- 4) X-Lite přihlášeným jako uživatel 100 vytočte číslo 999 pro přihlášení agenta. Při spojení budete dotázáni na identifikační číslo, heslo agenta a příčinný extension. V našem případě ID agenta je 007 a heslo 1234. Pro extension zadáme 100.
- 5) Druhým klientem X-Lite vytočíme číslo 112, kterým se připojíme do volací fronty. Uživatel přihlášený jako agent by měl začít vyzvánět. Při zvednutí hovoru by mělo nastat spojení.

Kontrolní otázky:

- 1) K čemu a kde se využívá signalizačního protokolu SIP?
- 2) Jaké operační systémy podporuje ústředna Asterisk?
- 3) K čemu slouží volací fronty?
- 4) Co je X-Lite?

PŘÍLOHA 2: UKÁZKY KONFIGURAČNÍCH SOUBORŮ

sip.conf

```
[general]
port = 5060
context = sip
disallow = all
allow = ulaw

[3001]
type = friend
username = 3001
secret = 1234
host = dynamic
callerid = Agent_1 <3001>

[3002]
type = friend
username = 3002
secret = 1234
host = dynamic
callerid = Agent_2 <3002>

[3003]
type = friend
username = 3003
secret = 1234
host = dynamic
callerid = Agent_3 <3003>

[3004]
type = friend
username = 3004
secret = 1234
host = dynamic
callerid = Agent_4 <3004>
```

agents.conf

```
[Agents]
ackcall = no
musiconhold = default
autologoff = 15

group = 1
agent => 1001,1234,Agent_1
agent => 1002,1234,Agent_2
agent => 1003,1234,Agent_3
agent => 1004,1234,Agent_4
```

queues.conf

```
[queue]
musiconhold = default
strategy = leastrecent
timeout = 15
retry = 5
wrapuptime = 0
maxlen = 0
announce-frequency = 0
announce-holdtime = no

member => Agent/@1
```

musiconhold.conf

```
[default]
mode = files
directory = /var/lib/asterisk/moh
random = yes
```

extensions.conf

```
[general]
static = yes
writeprotect = yes

[sip]
;call centrum
exten => 2100,1,Answer
exten => 2100,2,Ringing
exten => 2100,3,Wait(5)
exten => 2100,4,Queue(queue)
exten => 2100,5,Hangup

;SIP clients
exten => 3001,1,Dial(SIP/3001)
exten => 3002,1,Dial(SIP/3002)
.
.
.
exten => 3020,1,Dial(SIP/3020)

;log in/out agents
exten => 2001,1,AgentCallbackLogin(||${CALLERID(NUM)}@operator)
exten => 2002,1,AgentCallbackLogin(||@operator)

[operator]
exten => 3001,1,Dial(SIP/3001)
exten => 3002,1,Dial(SIP/3002)
exten => 3003,1,Dial(SIP/3003)
exten => 3004,1,Dial(SIP/3004)
```