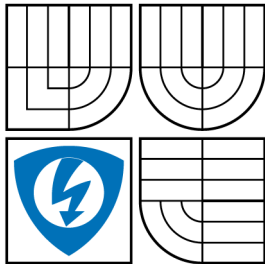


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

SPRÁVA A KONFIGURACE VOIP ÚSTŘEDNY ASTERISK ADMINISTRATION AND CONFIGURATION OF ASTERISK VOIP EXCHANGE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

BC. TOMÁŠ BINDER

VEDOUcí PRÁCE
SUPERVISOR

ING. PETR ČÍKA

BRNO 2008

ABSTRAKT

Diplomová práce se zabývá VoIPov softwarem ústřednou Asterisk. Jsou zde popsány její možnosti a způsoby její konfigurace. Zvláštní pozornost je věnována protokolu SIP, který popsán v jedné z kapitol. V rámci diplomové práce byl vytvořen číslovací plán, který demonstruje jakým způsobem se číslovací plány vytváří. Jsou v něm demonstrovány následující služby: hlasová schránka, konference, interaktivní hlasový automat a fronta hovorů. V diplomové práci jsou popsány konfigurační soubory, pomocí kterých se ústředna konfiguruje. Závěrem práce jsou uvedeny tři laboratorní úlohy pro potřeby předmětu Multimediální služby. Úlohy by měly seznámit studenty s vytvářením SIPových účtů na ústředně, jejich vzájemným spojováním, definicí interaktivního hlasového automatu a vytvořením call centra.

KLÍČOVÁ SLOVA

Asterisk, SIP, VoIP, PBX, ústředna

ABSTRACT

Abstract in English This diploma dissertation is dealing with the VoIP software exchange Asterisk. In the dissertation there are described its abilities and possible ways of its configuration. Special attention is given to the signalling protocol SIP, which is described in one of the chapters. Within this dissertation a dial plan, which demonstrates the technique of dial plan creating, was created. Within the boundaries of the dialplan following services could be found: a voicemail, conference, Interactive Voice Response and call queues. Configuration files, with the help of which the exchange is configured, are described in my dissertation as well. Finally, three laboratory assignments for purposes of the subject Multimedia Services are mentioned. Their main aim is to familiarise students with the creation of SIP accounts in the exchange, their mutual connections, defining the Interactive Voice Response and forming a new call centre.

KEYWORDS

Asterisk, SIP, VoIP, PBX, exchange, softswitch

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Správa a konfigurace VoIP ústředny Asterisk “ jsem vypracoval samostatně pod vedením vedoucího diplomové 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové 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)

OBSAH

Úvod	11
1 Co je to Asterisk	12
1.1 Základní části Asterisku	13
2 Protokol SIP	15
2.1 Prvky sítě SIP	15
2.2 Funkce protokolu SIP	16
2.2.1 Rozeznávání adres	17
2.2.2 Funkce související s relací	17
2.2.3 Funkce nesouvisející s relací	19
2.2.4 Příklady SIP zpráv	20
3 Konfigurace SIP účtů	22
3.1 Všeobecné nastavení (general)	22
3.2 Nastavení klientů	22
3.3 Příklad sip.conf	23
3.4 Spojení dvou Asterisků pomocí protokolu SIP	24
3.5 sip.conf pro demonstrační číslovací plán	25
4 Číslovací plány	26
5 Služby pro call centra	30
5.1 Systém front hovorů	31
5.2 Agenti	32
5.3 Parkování a přenos hovoru	34
6 Ostatní služby	36
6.1 Soubor voicemail.conf	36
6.2 Soubor musiconhold.conf	37
6.3 Soubor followme.conf	37
6.4 Soubor meetme.conf	38
7 Demonstrační číslovací plán	39
7.1 Kontext from-internal	41
7.2 Kontext ivr-hlavni	44
7.3 Kontext ivr-VM	45
7.4 Kontexty pro fronty	45

7.5	Kontext Callback-agents	46
7.6	Kontext from-sip-external	46
7.7	Kontext from-sip-external-ast-sec	47
7.8	Kontext outbound-routes	47
7.9	Kontext hledej	47
7.10	Makra	47
	7.10.1 macro-dialout-trunk	47
	7.10.2 macro-volani-na-klapku-vm	48
	7.10.3 macro-systemrecording	49
8	Úloha 1 - SIP	50
8.1	Připojení k Asterisku	50
8.2	Asterisk CLI	50
8.3	Konfigurace softwarového telefonu X-Lite	50
8.4	Postup	51
8.5	Zachycení SIP zpráv	51
9	Úloha 2 - IVR	53
9.1	Interactive voice response	53
9.2	Postup	53
10	Úloha 3 - Call Centrum	57
10.1	Postup	57
11	Závěr	59
	Literatura	60

SEZNAM OBRÁZKŮ

5.1	Funkce call centra	30
7.1	Schema testovacího pracoviště	39
9.1	IVR menu pro call centrum	55

SEZNAM TABULEK

1.1	Používané kodeky	12
2.1	Typy SIPových metod	15
2.2	Třídy SIPových odezev	15
2.3	Popis sloupců v příkladech komunikace	20
7.1	Globalní proměnné	40
7.2	Definované klapky v kontextu <code>from-internal</code>	43
7.3	Služby v kontextu <code>from-internal</code>	44
7.4	Číselné volby definované v <code>ivr-hlavni</code>	45
7.5	Číselné volby definované v <code>ivr-VM</code>	45
10.1	Tabulka front a agentů	58

ÚVOD

V současné době se začíná čím dál tím více využívat pro telekomunikační služby protokolu IP. Toto řešení je často označováno anglickým termínem Voice over IP, VoIP. Pro služby VoIP je možno využít internet, intranet nebo libovolného datového spojení s dostatečnou šířkou pásma. Důležitým požadavkem VoIPu, kladeným na datovou síť, je zajištění kvality služeb (QoS). Pro potřeby signalizace se současné době nejvíce využívá protokol SIP.

V této diplomové práci bude představena telefonní ústředna Asterisk, která je schopna nahradit klasické pobočkové ústředny. Za tímto účelem bude vytvořen demonstrační číslovací plán, který se bude snažit ukázat jakým způsobem se číslovací plány tvoří. Budou demonstrovány služby jako jsou hlasová schránka, konference, interaktivní hlasový automat a fronta hovorů.

Dalším cílem diplomové práce bude vytvořit laboratorní úlohy pro potřeby předmětu Multimediální služby. Úlohy by měli posloužit studentům ke získání základního přehledu vlastností a možností ústředny Asterisk.

Všechny konfigurace budou provedeny na linuxové distribuci trixbox CE 2.6.0.7 s Asteriskem verze 1.4.19. Konfigurace budou prováděny výhradně pomocí konfiguračních souborů.

1 CO JE TO ASTERISK

Asterisk je open source řešení pobočkové telefonní ústředny, které vytvořila firma Digium. Tato pobočková ústředna umožňuje připojit množství telefonů, které mohou telefonovat mezi sebou, spojit s veřejnou telefonní sítí a a využívat služby, které Asterisk nabízí. V této kapitole bylo čerpáno z pramenů [1] a [3].

Asterisk je šířen pod dvěma licencemi, jednou je GNU GPL a druhou je proprietární licence, která umožňuje využití uzavřených a patentovaných zdrojových kódů jako je např. G.729 kodek. Tato ústředna byla původně navržena pro operační systém Linux, ale nyní je možno jí provozovat na dalších unixových systémech (NetBSD, OpenBSD, FreeBSD, Mac OS X a Solaris).

Asterisk podporuje tyto protokoly

- IAX (Inter-Asterisk Exchange),
- H.323,
- SIP (Session Initiation Protocol),
- MGCP (Media Gateway Control Protocol),
- SCCP (Skinny Call Control Protocol).

Kodeky, které je schopen Asterisk používat jsou uvedeny v následující tab. 1.1.

Tab. 1.1: Používané kodeky

Kodek	Šířka pásma
G.711	64 kbps
G.726	16, 24 nebo 32 kbps
G.723.1	5,3 nebo 6,3 kbps
G.729A	8 kbps
GSM	13 kbps
iLBC	13,3 kbps nebo 15,2 kbps
Speex	Proměnná

PBX

Asterisk umožňuje vytvořit pobočkovou ústřednu (PBX), která má stejné vlastnosti jako klasická pobočková ústředna. Použitím Asterisku, telefonních karet a obyčejného PC je možno nahradit stávající PBX nebo jí doplnit o VoIP, hlasové schránky, konference a mnoho dalších služeb.

Interaktivní hlasový systém (IVR)

Asterisk pomocí svého IVR umožňuje uživatelům využívat databáze pomocí menu předem nahraných hlasových zpráv. Volající může s ústřednou komunikovat pomocí tónové volby, nahrávat odpovědi, dotazovat se databáze a využívat AGI skripty k provedení určitých úkolů.

Automatická obsluha

Obsluha zahrnuje pozdravy, hudbu při čekání (Music on hold) a předávání hlasových zpráv. Asterisk přehrává hudbu nebo připravené zprávy zákazníkům.

Konferenční most

Konferenční most je plně integrován do Asterisku a podporuje základní vlastnosti pro obchodní konference. Správce konference může vybrat, zda účastníkům konference bude povoleno pouze naslouchat nebo jim bude umožněno i mluvit. Pokud správce konference zavěsí budou odpojeni i ostatní uživatelé. Konference mohou mít zabezpečený přístup pomocí předurčeného PINu.

Media Server

Asterisk rozšiřuje možnosti existujících PBX a výchozích brán o vybrané vlastnosti PSTN nebo o IP protokoly. Asterisk může rozšířit vlastnosti a funkčnost využitím hlasové schránky, konferenčních služeb, front hovorů atd.

VoIP a protokolová brána

Asterisk podporuje jak tradiční TDM tak i VoIP protokoly což z něj umožňuje udělat bránu mezi různými typy kanálů. Použitím Asterisku je možno udělat i mnohem sofistikovanější výchozí brány s redundantními linkami. Např. konvertor z MGCP na SIP se záložní PRI linkou, když je SIP trunk nedostupný.

1.1 Základní části Asterisku

Asterisk je tvořen několika částmi jako jsou moduly, konfigurace, zvukové soubory. Soubor **asterisk.conf** obsahuje základní nastavení Asterisku

- nastavení pro vzdálenou konzoli k Asterisku (sekce **files**),
- umístění různých adresářů, které Asterisk používá (sekce **directories**),
- výchozí nastavení příkazové řádky (sekce **options**).

Umístění jednotlivých částí Asterisku je nastaveno v sekci `directories`. Dále jsou uvedeny jednotlivé volby sekce `directories` a jejich standardní nastavení.

- `astetcdir` => `/etc/asterisk` - konfigurační soubory,
- `astmoddir` => `/usr/lib/asterisk/modules` - moduly Asterisku (např. `ko-deky`, `chan_iax2.so`, `app_dial.so` atd.),
- `astvarlibdir` => `/var/lib/asterisk` - zvuky (`*.gsm`), MOH nahrávky,
- `astagidir` => `/var/lib/asterisk/agi-bin` - AGI skripty,
- `astspooldir` => `/var/spool/asterisk` - záznamy (`voicemail`, `meetme`, `record`),
- `astrundir` => `/var/run` - soubor s PID, které identifikuje běžící Asterisk,
- `astlogdir` => `/var/log/asterisk` - logovací soubory Asterisku.

2 PROTOKOL SIP

SIP (Session Initiation Protocol) je textově založený signalizační protokol určený pro multimediální komunikaci peer-to-peer (účastníci komunikace jsou si rovni). Protokol SIP však používá transakce typu klient-server. Klient generuje požadavky a server na ně odpovídá vygenerováním odpovědi. V této kapitole bylo čerpáno z pramene [4]. Protokol SIP je specifikován v RFC 3261 [6]. Typy nejdůležitějších SIP požadavků (známých též jako metody) jsou uvedeny v následující tab. 2.1.

Tab. 2.1: Typy SIPových metod

SIP metoda	Popis
INVITE	Sestavení relace
ACK	Konečné potvrzení INVITE
BYE	Ukončení existující relace
CANCEL	Zrušení dosud neseštavené relace
REGISTER	Registrace uživatele
OPTIONS	Dotaz na možnosti a nastavení

Odpovědi protokolu SIP jsou číselné. Tyto odpovědi jsou rozděleny do šesti tříd, které jsou uvedeny v tab. 2.2.

Tab. 2.2: Třídy SIPových odezev

Třída	Popis
1xx	Informativní - Požadavek se zpracovává, ale zatím není kompletní
2xx	Úspěch - Požadavek byl úspěšný
3xx	Přesměrování - Požadavek by měl být vyzkoušen jinde
4xx	Chyba klienta - Požadavek nebyl úspěšný (chyba v požadavku)
5xx	Chyba serveru - Požadavek nebyl úspěšný (chyba příjemce)
6xx	Komplexní selhání - Požadavek nebyl úspěšný

2.1 Prvky sítě SIP

V sítích využívajících protokol SIP jsou tři hlavní prvky: uživatelští agenti, servery, služby pro vyhledávání.

Uživatelští agenti

Uživatelští agenti (UA - User Agent) jsou koncovými body v SIPových sítích. Tito agenti vytvářejí požadavky k sestavení multimediálního toku. Uživatelský agent

může být hardwarový nebo softwarový SIPový telefon nebo výchozí brána do jiné sítě (např. PSTN). Uživatelský agent-klient (UAC - User Agent Client) je část UA, která generuje požadavky, zatímco uživatelský agent-server (UAS - User Agent Server) je část UA, která generuje odezvy na obdržené požadavky. Každé koncové zařízení (UA) podporující protokol SIP obsahuje jak UAC tak i UAS.

Servery

Servery jsou zprostředkovávající zařízení, které asistují uživatelským agentům při sestavování SIPové relace a jiných činnostech. Existují tři druhy SIPových serverů.

- SIP proxy - přijímá SIPové požadavky od UA nebo jiné proxy a přeposílá je dál do sítě.
- Redirect server - přijímá SIPové požadavky od UA nebo proxy a vrací odezvu přesměrování (3xx), tím oznamuje, kde by měl být požadavek znovu vyzkoušen.
- Registrar server - přijímá SIP požadavky o registraci a aktualizuje informace o UA v lokalizačních službě nebo jiné databázi.

Všechny tři druhy serverů slouží pouze pro potřeby signalizace. Nemohou generovat žádný mediální multimediální tok ani žádné SIPové požadavky (pouze zastupují UA).

Lokalizační služba

Lokalizační služba je v podstatě databáze, která obsahuje informace o uživatelských agentech (URI, IP adresy, skripty, možnosti a jejich nastavení). Může též obsahovat směrovací informace o SIPové síti. UA nespolupracuje přímo s lokalizační službou, nýbrž pomocí proxy, redirect nebo registrar serveru.

2.2 Funkce protokolu SIP

Zde budou vysvětleny nejdůležitější funkce protokolu SIP, jedná se o rozeznávání adres, sestavení spojení, vyjednávání o multimediálním toku, modifikace relace, ukončení relace, mobilita a autentizační výzvy.

2.2.1 Rozeznávání adres

Rozeznávání adres obvykle začíná URI (Uniform Resource Identifier) a končí uživatelským jménem na nějaké IP adrese. Toto rozeznávání může provádět jak uživatelský klient tak i uživatelský server. Rozeznávání adres může zahrnovat následující kroky:

- DNS NAPTR dotaz k určení transportního protokolu (UDP, TCP, SCTP),
- DNS SRV vyhledávání, které určí jméno serveru a číslo portu,
- DNS A vyhledávání, které určí IP adresu serveru,
- ENUM vyhledávání telefonního čísla,
- pokud se směřuje k proxy serveru v doméně uživatele provede se i vyhledávání v lokalizační službě.

Tento proces rozeznávání adresy probíhá pouze jednou na počátku relace. Výsledky tohoto procesu jsou uloženy a později použity pro další požadavky mezi uživatelskými agenty.

2.2.2 Funkce související s relací

Funkce protokolu SIP zahrnují sestavení relace, události během sestavené relace a ukončení relace.

Sestavení relace

K sestavení relace mezi dvěma uživatelskými agenty používá protokol SIP zprávu INVITE. Tělo zprávy obvykle obsahuje typ relace, které si přeje inicializující agent sestavit. SIPový UAC vyplní ve zprávě tyto tři důležité položky:

- To: - komu je voláno,
- From: - od koho je voláno,
- Call-ID: - jedinečný identifikátor pro tuto relaci (hovor).

Kombinace těchto tří hodnot jedinečně identifikuje hovor a během hovoru se nikdy nemění. Někdy jsou tyto hodnoty označovány jako dialog. Sestavení relace obsahuje výměnu tří zpráv INVITE/200/ACK (označuje se také jako three-way handshake), tato kombinace představuje úspěšné sestavení relace. Neúspěšný pokus o sestavení relace vypadá takto INVITE/4xx, 5xx nebo 6xx/ACK.

Pokud je multimediální tok již sestaven neprobíhá výměna SIPových zpráv, dokud si jedna ze stran nepřeje relaci modifikovat (re-INVITE) nebo ukončit (BYE). Příklad jak vypadá konkrétní sestavení relace je uvedena v podkapitole 2.2.4.

Vyjednávání o multimediálního toku

Vyjednávání o multimediálním toku je součástí sestavení relace mezi dvěma UA. Protokol SIP sám o sobě nepodporuje vyjednávání o multimediálním toku. Toto vyjednávání se provádí pomocí protokolu pro popis relace (Session Description Protocol - SDP). Tento textově orientovaný protokol je definován v RFC 4566 [7]. Zpráva protokolu SDP se připojuje k hlavičce zprávy INVITE, pokud volající nastavuje parametry toku, nebo k hlavičce zprávy 200 OK, pokud volající přenechává nastavení parametrů na volaném. SDP umožňuje specifikovat typ multimediálního toku, jeho kodeky, IP adresu a transportní port. Druhá strana porovná požadované nastavení multimediálního toku se svými možnostmi. Následně výsledky svého porovnání zašle zpět inicializujícímu UA.

Následuje komentovaná ukázka SDP, kde jsou nabízeny tři audio kodeky (GSM, a-law a iLBC) a jeden video kodek (H263).

```
v=0 ; verze SDP
o=root 23747 23747 IN IP4 147.229.xxx.xxx ; vlastník a identifikátor relace
s=session ; jméno relace
t=0 0 ; čas jaký je relace aktivní
m=audio 14886 RTP/AVP 3 8 97 101 ; typ toku, transportní port
; transport. prot., seznam formátů
a=rtpmap:3 GSM/8000 ; typy payloadu, jména kodeků
a=rtpmap:8 PCMA/8000 ; /vzorkování
a=rtpmap:97 iLBC/8000
a=fmtp:97 mode=30 ; specifické parametry kodeku iLBC
a=rtpmap:101 telephone-event/8000 ; DTMF pomocí RTP dle rfc2833
a=fmtp:101 0-16 ; specif. parametry přenosu DTMF
m=video 19332 RTP/AVP 34 ; další typ mediální toku
a=rtpmap:34 H263/90000 ; typ payloadu, jméno kodeku
```

Modifikace relace

Pokud už relace byla jednou navázána pomocí sekvence INVITE/200/ACK, může být tato relace upravena pomocí další sekvence INVITE/200/ACK, která bývá nazývána re-INVITE. Tato modifikace může proběhnout až poté co předchozí INVITE bylo dokončeno pomocí zprávy ACK. Re-INVITE může provést jakýkoliv účastník používající

stejné parametry To, From a Call-ID. Pokud je re-INVITE úspěšný je nahrazen původní INVITE SDP za re-INVITE SDP, ale když je re-INVITE z jakéhokoliv důvodu neúspěšný bude se nadále používat původní INVITE SDP, dokud není relace ukončena.

Ukončení a zrušení relace

Ukončení relace nastává pokud jeden ze spojených UA zašle zprávu BYE, která se odkazuje na existující SIP dialog. Zrušení relace nastává, pokud UA ukončuje relaci, která ještě nebyla kompletně sestavena (nebyla potvrzena pomocí ACK), toto se realizuje zasláním zprávy CANCEL. Tedy vznikající relace se ruší pomocí metody CANCEL, zatímco relace sestavená se ukončuje metodou BYE. Příklady jak vypadá konkrétní ukončení a zrušení relace jsou uvedeny v podkapitole 2.2.4.

2.2.3 Funkce nesouvisející s relací

Mobilita

Mobilita je zajištěna tím, že každý AU se zaregistruje pomocí metody REGISTER na registrar serveru. V této zprávě UA vyplní položku Contact do které uvede URI, na kterém si přeje přijímat hovory. Registrar server poté aktualizuje informace v lokalizační službě, tedy sdělí lokalizačnímu serveru uživatelské jméno, IP adresu a port, kde může být UA dostižen. Aktualizace lokalizačního serveru není prováděna pomocí SIP protokolu. Po úspěšné registraci a aktualizaci lokalizační služby odpovídá registrar server zprávou 200 OK. Položek v Contact v metodě REGISTER může být několik.

Autentizační výzvy

SIP podporuje dva typy autentizačních výzev: UA s UA a UA se serverem. SIP podporuje několik autentizačních schémat převzatých z protokolu HTTP. Nejpoužívanější autentizační schéma je Digest, které spoléhá na výzvu, odezvu a sdílené tajemství mezi UA a registrar serverem nebo druhým UA.

Sdílené tajemství je obvykle šifrované uživatelské jméno a heslo. Typická výměna zpráv INVITE/401 Authentication Required/ACK. Z této výměny zjistí inicializující UA, že druhá strana vyžaduje autentizaci. Tento UA tedy zašle novou zprávu INVITE s vyplněnou položkou Authentication. Pokud tato položka obsahuje správné informace pokračuje se obvyklým způsobem. Jinak je odezva opět 401 Authentication Required. Další odezvou, která vyžaduje autentizaci je 407 Proxy Authentication Required. To oznamuje UA, že má uvést a vyplnit položku Proxy-AuthORIZATION.

2.2.4 Příklady SIP zpráv

Následující výměny zpráv mezi softwarovým telefonem X-lite a Asteriskem s demonstračním číslovacím plánem, byly zachyceny na klientském počítači pomocí Wiresharku. Vysvětlení jednotlivých sloupců v příkladech komunikace je uveden v následující tab. 2.3. V příkladech nejsou uváděny IP adresy.

Tab. 2.3: Popis sloupců v příkladech komunikace

Označení	Význam
N	Pořadí zprávy
S	Zdroj
D	Cíl
Prot.	Protokol
Info	Obsah zprávy

Registrace X-litu do Asterisku

Následující výpis zpráv ukazuje registraci softwarového telefonu X-lite (K) do Asterisku (A). X-lite zašle zprávu REGISTER, která neobsahuje autentizaci. Asterisk oznamuje zprávou 401 Unauthorized klientovy, že vyžaduje autorizaci. Klient do druhé zprávy REGISTER doplní autentizační údaje. Poslední zprávou je potvrzení Asterisku, že autorizační údaje byly v pořádku.

```
N  S  D  Prot.  Info
1  K  A  SIP    Request: REGISTER sip:A;transport=UDP
2  A  K  SIP    Status: 100 Trying
3  A  K  SIP    Status: 401 Unauthorized
4  K  A  SIP    Request: REGISTER sip:A;transport=UDP
5  A  K  SIP    Status: 100 Trying
6  A  K  SIP    Status: 200 OK
```

Zprávy vyměněné při sestavování hovoru

Nejprve bude ukázáno sestavování hovoru, který nebyl volaným přijat. Následně bude uveden příklad zpráv spojeného hovoru.

Klient (K) požádal o spojení s klapkou 15 (INVITE+SDP). Asterisk (A) požádal klienta o autentizaci. Klient doplnil do zprávy autorizační údaje. Asterisk začal hovor spojovat (zpráva 100 Trying a 180 Ringing). Volaný klient začal vyzvánět. Volaný hovor nezvedl. Volající klient hovor zavěsil (zpráva CANCEL). Asterisk požadavek

na spojení zrušil (zprávy 487 Request Terminated a 200 OK). Klient potvrdil, že je s tím srozuměn (ACK).

```
N S D Prot. Info
1 K A SIP/SDP Request: INVITE sip:15@A, with session desc.
2 A K SIP Status: 407 Proxy Authentication Required
3 K A SIP Request: ACK sip:15@A
4 K A SIP/SDP Request: INVITE sip:15@A, with session desc.
5 A K SIP Status: 100 Trying
6 A K SIP Status: 180 Ringing
7 K A SIP Request: CANCEL sip:15@A
8 A K SIP Status: 487 Request Terminated
9 A K SIP Status: 200 OK
10 K A SIP Request: ACK sip:15@A
```

Počátek scénáře je shodný s předchozím příkladem, ale volaný tentokrát hovor přijal. Hovor proběhl. Poté volaný zavěsil. Volající obdržel zprávu BYE. Volající toto potvrdil zprávou 200 OK.

```
N S D Proto. Info
1 K A SIP/SDP Request: INVITE sip:15@A, with session desc.
2 A K SIP Status: 407 Proxy Authentication Required
3 K A SIP Request: ACK sip:15@A
4 K A SIP/SDP Request: INVITE sip:15@A, with session desc.
5 A K SIP Status: 100 Trying
6 A K SIP Status: 180 Ringing
7 A K SIP/SDP Status: 200 OK, with session description
8 K A SIP Request: ACK sip:15@A
-----HOVOR-----
9 A K SIP Request: BYE sip:11@K:55140
10 K A SIP Status: 200 OK
```

3 KONFIGURACE SIP ÚČTŮ

Soubor **sip.conf** obsahuje nastavení SIPových kanálů. Klienti musí být definováni v tomto souboru, aby mohli volat nebo být voláni. Soubor obsahuje obecné nastavení (IP adresa, port atd.). Následující sekce určují jednotlivé klienty. Jména klientů jsou uzavřeny v hranatých závorkách a následují jejich volby (uživatelské jméno, heslo, DTMF atd.). Při popisu konfigurace bylo čerpáno z pramenů [1] a [2].

3.1 Všeobecné nastavení (general)

V sekci **general** se definují všeobecné údaje, které jsou platné pro všechny definice SIPových připojení, pokud není v definici účtu uvedeno jinak. V sekci **general** mohou být definovány následující volby.

context=jmeno_con - nastaví **jmeno_con** jako defaultní kontext, tedy kontext kam budou všechny hovory umístovány, pokud není uvedeno jinak (kontext je možné nastavit každému uživateli individuálně).

port=5060 - UDP port na kterém Asterisk očekává příchozí SIP spojení.

bindaddr=0.0.0.0 - IP adresa na které Asterisk naslouchá. 0.0.0.0 znamená naslouchání na všech dostupných adresách.

srvlookup=yes - povolí dotaz DNS SRV pro odchozí volání. Pokud je volba nastavena na "no" není možno určovat adresy pomocí doménových jmen.

useragent=Ast_Ustredna - tato hodnota bude doplňována do SIP hlavičky do pole **useragent**. Výchozí hodnota je **Asterisk**.

videosupport=yes - zapne podporu videa.

nat=yes - Asterisk ignoruje IP adresy v SIP a SDP hlavičkách a odpovídá na IP adresu a port uvedené v IP hlavičce.

allow=alaw - povolí kodek alaw.

3.2 Nastavení klientů

type= - jeho volby jsou

peer - zařízení, které přijímá hovory od Asterisku,

user - zařízení, které volá přes Asterisk,

friend - zařízení, se chová jak peer tak i jako user (hovory inicializuje i přijímá).

secret=heslo - nastaví klientovy heslo.

host=dynamic - nastaví IP adresu nebo DNS jméno zařízení. V DHCP sítích může být nastaveno "dynamic".

defaultip=192.168.0.10 - toto nastavení se používá pokud je host nastaven na "dynamic". Asterisk se pokusí spojit hovor na uvedenou IP adresu, pokud je obdržen hovor pro SIP klienta, který není přihlášen.

context=uziv-kontext - definice kontextu pouze pro daného SIP klienta.

canreinvite=no - zakáže měnit parametry multimediálního toku během probíhajícího hovoru (INVITE s SDP).

mailbox=11@internal - povolí zjištění obsahu hlasové schránky pomocí metody SUBSCRIBE.

insecure=invite - Asterisk nebude vyžadovat autentizaci pro příchozí INVITE.

3.3 Příklad sip.conf

```
[general]
port=5060
bindaddr=0.0.0.0
disallow=all
allow=alaw
allow=gsm
allow=h263

[ast-sec]
type=friend
secret=well
context=from-sip-external-ast-sec
host=dynamic
insecure=invite

[11]
type=friend
secret=asdf
```

```
callerid=Drobek <11>
host=dynamic
context=from-internal
canreinvite=yes
mailbox=11@internal
```

3.4 Spojení dvou Asterisků pomocí protokolu SIP

Pokud máme více ústředen a chceme mezi nimi předávat hovory, případně se připojit k poskytovateli, můžeme vytvořit trunk, což znamená vytvořit, případně nechat vytvořit účet na jedné ústředně a druhou ústřednou se na tento účet přihlásit pomocí volby **register** v sekci **general** v souboru **sip.conf**. Tato volba oznámí druhé ústředně kam má směřovat hovory určené pro první ústřednu. Tedy řečeno slovy "Pokud máš pro mě hovor, pošli mi ho na moji IP adresu". Parametr **register** vypadá následovně

```
register => username:secret@poskytovatel.cz.
```

Obsah souboru **sip.conf** na první ústředně

```
[general]
register => toronto:welcome@192.168.1.101/osaka

[osaka]
type=friend
secret=welcome
context=osaka_incoming
host=dynamic
disallow=all
allow=ulaw
```

Obsah souboru **sip.conf** na druhé ústředně

```
[general]
register => osaka:welcome@192.168.2.202/toronto
[toronto]
type=friend
secret=welcome
context=toronto_incoming
host=dynamic
disallow=all
allow=ulaw
```

V číslovacím plánu první ústředny je možno definovat

```
[oslnovice_calls]
exten => _1XXX,n,Dial(SIP/osaka/${EXTEN})
```

V číslovacím plánu druhé ústředny je možno definovat

```
[brno_calls]
exten => _2XXX,n,Dial(SIP/toronto/${EXTEN})
```

3.5 sip.conf pro demonstrační číslovací plán

Pro potřeby demonstračního číslovacího plánu je v souboru **sip.conf** nakonfigurováno devět uživatelů (11 až 19), účet pro přihlášení druhého Asterisku (ast-sec) a účet určený pro vytáčení poskytovatele (trunk_094). V konfiguraci jsou uvedeny i dvě registrace. První je k poskytovateli a druhá je k druhé ústředně ast-sec.

4 ČÍSLOVACÍ PLÁNY

Číslovací plán (dialplan) je pravým srdcem Asterisku. Je to sada instrukcí, jakým způsobem se mají jednotlivé multimediální toky spouštět a řídit. Vytváření číslovacího plánu je jako psaní kódu ve speciálním skriptovacím jazyce. Asterisk je schopen pracovat s proměnnými, operátory, makry atd. Tyto "skripty" zajišťují kontrolu příchozích a odchozích hovorů a jejich směrování. Číslovací plán je uložen v souboru **extensions.conf** a je rozdělen do několika sekcí, které obsahují buď statické definice nebo spustitelné číslovací plány, kterým se říká **kontexty**. Statické definice se nacházejí ve dvou sekcích **general** a **globals**. Při popisu číslovacích plánů bylo čerpáno z literatury [1].

Jména jednotlivých kontextů určuje správce. V konfiguraci se vyskytují další speciální kontexty, které začínají řetězcem **macro-**, které je možno volat z celého číslovacího plánu. Každá klapka je složena z klíčového slova **exten**, znaků "=>" a dále následuje

- jméno (nebo číslo) klapky,
- priorita,
- aplikace, která provádí nějakou akci.

Celá definice klapky vypadá takto:

```
exten => 765,1,application()
```

Statické definice

Statické definice se nacházejí v sekcích **general** a **globals**. Sekce **general** obsahuje nastavení, která se vztahují přímo k souboru **extensions.conf**. Např. zda je možno soubor přepisovat (**writeprotect**), zda se mají mazat globální proměnné při reloadu (**clearglobalvars**) atd. Sekce **globals** obsahuje globální proměnné, které mohou být následně využívány v celém číslovacím plánu.

Priority

Pokud budeme chtít provést s hovorem několik operací. Je nutné určit, která operace se provede jako první, která jako druhá atd., k tomu slouží priority. Klapka by měla začínat prioritou 1, pokud by to bylo číslo vyšší Asterisk tyto operace neprovede.

```
exten => 765,1,Answer()  
exten => 765,2,Playback(hello-world)  
exten => 765,3,Hangup()
```

Asterisk umožňuje použít prioritu `n`, což znamená následující priorita. Předchozí příklad by tedy vypadal následovně.

```
exten => 765,1,Answer()  
exten => 765,n,Playback(hello-world)  
exten => 765,n,Hangup()
```

Jednotlivé priority mohou být pojmenovány, jak je vidět dále.

```
exten => 765,n(jmeno),Playback(hello-world)
```

Aplikace

Aplikace slouží k provedení požadované operace s kanálem např. přijme DTMF tón, přehraje zvuk, zavěsí hovor atd. Aplikace jako `Answer()` nebo `Hangup()` nepotřebují další informace k provedení jejich operace. Jiné aplikace tyto dodatečné informace potřebují a to jsou argumenty aplikace např. `Playback(hello-world)`, kde `hello-world` je argumentem.

Syntaxe vzoru čísel

Při vyhledávání vzorů používáme písmena, která reprezentují skupiny čísel. Tyto výrazy vždy začínají symbolem ”_”. Pokud tento znak bude chybět Asterisk si bude myslet, že se jedná o název klapky. Po podtržítku lze využít jeden z následujících znaků:

- `X` vybere jakoukoliv číslici od 0 do 9,
- `Z` vybere jakoukoliv číslici od 1 do 9,
- `N` vybere jakoukoliv číslici od 2 do 9,
- `[26-9]` vybere 2 nebo jakoukoliv číslici z rozsahu 6-9,
- `.` vybere jeden nebo více znaků (wildcard).

Takto by vypadala klapka pro 9 a jakékoliv devítimístné číslo, které by bylo při vytočení řečeno do kanálu (včetně 9):

```
exten => _9XXXXXXXX,1,SayDigits(${EXTEN})
```


Výrazy a proměnné

Výrazy jsou kombinace proměnných, operátorů a hodnot. Výrazy mohou testovat proměnné, spojovat textové řetězce nebo provádět matematické výpočty. Výrazy vždy začínají dolarem a otevřenou hranatou závorkou a končí zavřením hranaté závorky, jak je vidět níže:

```
$(výraz]
```

Proměnné začínají dolarem a otevřenou složenou závorkou viz níže:

```
${proměnná}
```

Příkladem může být přičtení pěti k proměnné `POCET` viz níže:

```
exten => 898,1,Set(POCET = 6)
exten => 898,n,Set(VYSLEDEK =${${POCET} + 5})
exten => 898,n,SayNumber(${VYSLEDEK})
```

Operátory

Asterisk obsahuje tyto skupiny operátorů:

- boolean operátory, které určují pravdivost tvrzení

`vyraz1 | vyraz2` - nebo

`vyraz1 & vyraz2` - a zároveň

`vyraz2 [=, >, >=, <, <=, !=] výraz2` - porovnání textových řetězců
nebo celých čísel

- matematické operátory

`vyraz1 [+, -] vyraz2` - sčítání, odčítání

`vyraz1 [*, /, %] vyraz2` - násobení, celočíselné dělení, zbytek po celočíselném dělení

- Operátory regulárních výrazů

`vyraz1 : vyraz2` - porovnání `vyraz1` s `vyrazem2`, který musí být regulárním výrazem.

Asteriskový parser je poměrně jednoduchý, je třeba dávat alespoň jednu mezeru mezi operátor a hodnotu. Toto přiřazení do proměnné `TEST` bude bráno jako textový řetězec `"5+1"`

```
exten => 765,1,Set(TEST=${5+1})
```

Pokud ale bude použita před a za operátorem mezera, výsledek v proměnné TEST bude 6:

```
exten => 765,1,Set(TEST=${5 + 1})
```

Spojování textových řetězců se provádí pouhým spojením výrazů, jak je vidět níže:

```
exten => 765,1,Set(TEST=retez)
```

```
exten => 765,1,Set(${pred${TEST}za})
```

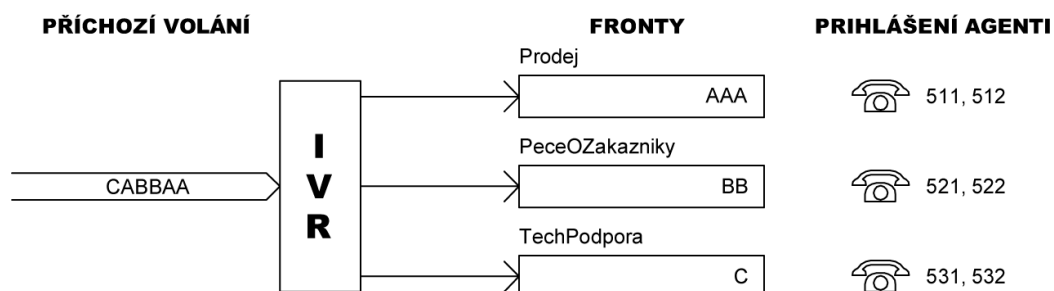
Výsledkem by byl tento textový řetězec "predretezza".

5 SLUŽBY PRO CALL CENTRA

Asterisk obsahuje tři základní funkční celky pro vytvoření call centra a to jsou [5] :

- systém front hovorů,
- agenti,
- interaktivní hlasová odezva (Interactive Voice Response - IVR).

Celý řetěz zpracování hovoru v call centru je vidět na následujícím obr. 5.1.



Obr. 5.1: Funkce call centra

Fronty hovorů

V Asterisku může být definována jedna nebo více front. Ty umožňují volajícím čekat, dokud nebude dostupný nějaký agent, kterým budou obslouženi. Fronty mohou být rozděleny podle typů požadavků (např. Prodej, Péče o zákazníky a Technická podpora).

Agenti

Agenti jsou osoby, které přijímají hovory z jednotlivých front, ke kterým jsou přiřazení.

IVR

IVR je mechanismus, který se snaží určit požadavky volajících a poté je přiřadit do vhodné fronty. IVR k rozlišení požadavků používá menu ze kterého si volající může vybrat, jakou službu požaduje.

Automatická distribuce hovoru (Automatic call distribution - ACD) spočívá ve směrování hovoru na speciální pseudokanál (chan_agent). Tedy pomocí IVR jsou hovory rozloženy do různých front, které obsluhují různí agenti. Pokud je agent přihlášen je hovor směrován k němu.

5.1 Systém front hovorů

Asterisk zajišťuje základní funkcionalitu call centra pomocí jeho systému front. Volající čeká ve frontě do té doby, dokud není volný agent, který jeho požadavek vyřídí. Fronty se konfigurují v souboru **queues.conf**, jeho důležité parametry jsou následující.

musiconhold => default - nastavuje hudbu, která bude přehrávána volajícímu při čekání.

strategy = - definuje jakým způsobem budou hovory distribuovány mezi agenty. Možnosti jsou následující

ringall - budou voláni všichni dostupní agenti,

leastrecent - bude volán agent, kterému nebyl hovor směřován nejdelší dobu,

fewestcalls - hovor bude směřován k agentovi s nejmenším počtem hovorů,

rrmemory - směřování hovoru bude cyklist po seznamu agentů.

joinempty = strict - nastavení určuje, že není možno volajícího vložit do fronty pokud jsou všichni agenti náležející frontě nedostupní.

leavewhenempty = strict - pokud je nějaký uživatel ve frontě a poslední agent obsluhující frontu se odhlásí, je uživatel z fronty okamžitě odstraněn a aplikace Queue() hovor předá následující prioritě.

member => typ/jmeno - určuje komu budou hovory z fronty směřovány. Členy mohou být kanály nebo agenti. Možnosti jsou následující:

Agent/cislo_agenta - členem se stane agent cislo_agenta,

Agent/@1 - členem se stane celá skupina 1,

SIP/uzivatel - členem se stane SIPový uživatel.

Příklad definice fronty z demonstračního plánu je vidět níže

```
[C_Prodej]
musicclass=default
strategy=ringall
context=C_Prodej
joinempty=strict
leavewhenempty=strict
member => Agent/@1
```

5.2 Agenti

Agenti jsou speciální uživatelé, kteří se mohou přihlašovat k Asterisku z libovolného účtu (např. SIP uživatel). Poté co se agent přihlásí, pomocí aplikace `AgentLogin()` nebo `AgentCallbackLogin()`, mohou na něj být směřovány hovory. Agenti se často využívají v kombinaci s frontami (viz podkapitola 5.1).

Agenty lze vytvářet a spravovat pomocí souboru `agents.conf`. Tento soubor se skládá ze sekce `general` a sekce `agents`, kde se definují jednotliví agenti a jejich volby. Stav jednotlivých agentů můžeme zjistit pomocí CLI příkazem `agent show`, který vypíše zda je agent přihlášen a případně na jaké klapce je dosažitelný. Následují vybrané volby ze souboru `agents.conf`. V sekci `general` je zajímavá pouze jedna následující volba

persistentmembers = yes - zajistí, že seznam agentů se bude ukládat do AstDB a to bude využito při opětovném startu. Argumenty mohou být `yes` a `no`.

Následující sekce `agents` může obsahovat následující volby.

maxlogintries = 2 - počet pokusů, kolikrát se může agent pokusit přihlásit. Standardní hodnota je 3.

autologoff = 15 - určuje, jak dlouho může agentovi zvonit přístroj než je považován za nedostupného a automaticky odhlášen. Argumentem této volby je čas v sekundách.

ackcall = yes - definuje, že agent bude muset při přihlašování, pomocí aplikace `AgentCallbackLogin()`, potvrzovat své volby tlačítkem "#".

wrapuptime = 5000 - určuje čas po jakém může být k agentovi směrován další hovor, pokud agent právě zavěsil. Čas se uvádí v milisekundách.

musiconhold => agent - určuje hudbu na pozadí, která bude hrát, pokud agent bude očekávat hovor.

group = 1 - definuje skupinu ke které agent náleží. Skupiny se definují pomocí celého čísla. Agent může náležet do více skupin, které jsou odděleny čárkou.

recordagentcalls = yes - určuje, zda budou hovory agentů nahrávány. Argumenty mohou být `yes` a `no`.

recordformat = gsm - nastavení formátu ve kterém budou soubory ukládány. Formáty mohou být `wav`, `gsm` nebo `wav49`. Standardní formát pro nahrávání je `wav`.

Dále následují definice jednotlivých agentů

agent => agent_id, heslo, jméno - toto nastevní vytvoří agenta s číslo **agent_id** s heslem a určeným jménem.

Konfigurace agentů pro potřeby demonstračního číslovacího plánu může vypadat následovně

```
[agents]
musiconhold = agent
autologoff=15
wrapuptime=5000
ackcall=no
; agenti pro linku Péče o zákazníky
group=1
agent => 511,1,Prodej_ag_1
agent => 512,1,Prodej_ag_2
; agenti pro linku Technické podpory
group=2
agent => 521,1,Pece0Zakazniky_ag_1
agent => 522,1,Pece0Zakazniky_ag_2
```

Aplikace `AgentLogin()` slouží k přihlášení agenta, jako parametr je možné uvést `agent_id`. Pokud číslo není uvedeno bude přihlašovaný vyzván k zadání `agent_id`. Dále musí volající zadat heslo. Ústředna oznámí, že agent byl úspěšně přihlášen a spustí definovanou hudbu. Jestliže bude použita aplikace `AgentCallbackLogin()` musí volající ještě zadat klapku na které bude dosažitelný. Ústředna ohlásí, že agent byl úspěšně přihlášen a zavěsí.

Definice klapek pro přihlášení agentů v demonstračním číslovacím plánu vypadá následovně

```
; trvalé přihlášení agenta
exten => *65,1,AgentLogin()
; přihlášení agenta pro zpětné volání
exten => *66,1,AgentCallbackLogin(,,$){CALLERID(num)}@Callback_agents)
; odhlášení agenta pro zpětné volání
exten => *67,1,AgentCallbackLogin(,@Callback_agents_nexist)
; odhlašovaný musí stisknout # při dotazu na klapku
```

V demonstračním číslovacím plánu je definován kontext `Callback_agents`, kde jsou vyjmenovány klapky na které bude hovor z fronty směrován.

```
; klapky pro AgentCallbackLogin
[Callback_agents]
exten => 11,1,Dial(SIP/11)
exten => 12,1,Dial(SIP/12)
```

5.3 Parkování a přenos hovoru

Aby mohl být hovor předán nebo zaparkován musí být spojen s možností přenosu hovoru. Toho lze docílit pomocí dvou příznaků aplikace `Dial()`:

- `t` - umožní volanému hovor předat,
- `T` - umožní volajícímu hovor předat.

Pokud je hovor spojen s těmito příznaky, je možno hovor předat tak, že se stiskne klávesa, která definována volbou `blindxfer`. Asterisk oznámí předání a uživatel zvolí klapku na kterou si přeje hovor předat. Parkování a přenos hovoru je možno nastavit v souboru `features.conf`.

Parkování hovoru

Parkování hovoru je speciálním případem předáním hovoru. Pokud se jeden z uživatelů rozhodne, že si přeje v hovoru pokračovat, ale z jiného místa (klapky), vytočí klapku, která je určena pro zaparkování hovoru. Asterisk mu následně oznámí, na jakou klapku, z rozsahu klapek určených pro parkování, hovor zaparkoval. Uživatel se přesune na jiné místo, kde použije nejbližší telefonní přístroj a vytočí klapku na kterou byl hovor zaparkován. Asterisk opět spojí čekajícího volajícího s novou klapkou. Parametry sekce `general` mají následující význam.

`parkext =>` - klapka, která se musí vytočit pro zaparkování hovoru.

`parkpos =>` - rozsah klapek kam může být hovor zaparkován.

`context =>` - kontext do kterého jsou hovoru zaparkovány.

`parkingtime =>` - určuje jak dlouho může hovor zůstat zaparkován. Pokud je určený čas překročen spojí ústředna hovor na původní klapku ze které byl hovor zaparkován. Čas je uveden v sekundách.

`featuredigittimeout` - definuje maximální čas mezi jednotlivými stisky kláves. Čas je uveden v milisekundách. Defaultní hodnota je 500.

V sekci `featuremap` se pomocí volby `blindxfer` určuje jaká klávesa (klávesy) slouží pro přenos hovoru. Opět následuje příklad pro demonstrační číslovací plán.

```
[general]
parkext => 800
parkpos => 801-820
parkingtime => 60
```

```
[featuremap]
blindxfer => ##
```

Posledním krokem je vložení kontextu `parkedcalls` do číslovacího plánu, jak je vidět níže

```
include => parkedcalls.
```


6 OSTATNÍ SLUŽBY

Další služby, které poskytuje Asterisk a jsou požity v demonstračním číslovací plánu jsou

- hlasová schránka - umožňuje volajícímu zanechat zprávu,
- hudba při držení hovoru - přehrává čekajícímu uživateli hudbu,
- "následuj mě" - snaží se lokalizovat uživatele,
- konference - umožňuje rozhovor několika volajících najednou.

Tyto funkce se konfigurují pomocí speciálních konfiguračních souborů a konkrétních parametrů spouštěných aplikací. Informace o konfiguračních souborech byly čerpány z pramenu [2].

6.1 Soubor `voicemail.conf`

Soubor `voicemail.conf` slouží ke konfiguraci hlasových schránek. Tento soubor je tvořen sekcemi `general`, `zonemessages` (nemusí být uvedena) a sekcí jednotlivých kontextů hlasových schránek. Volby sekce `general` mohou být následující

`format=wav|gsm` - určuje kodeky, které mohou být použity pro uložení hlasových zpráv.

`maxsilence=` - definuje, po kolika sekundách ticha je zpráva považována za ukončenou.

Následuje definice kontextu v hranatých závorkách a potom definice jednotlivých hlasových schránek. Definice jedné hlasové schránky vypadá takto

```
mailbox=heslo,jmeno[,email[,pager_email[,nastaveni]]].
```

Následuje příklad `voicemail.conf`, který je použit pro demonstrační číslovacím plánu.

```
[general]
format=wav|gsm
maxsilence=3
```

```
[internal]
11 => 1,Drobek
```

12 => 1,Trevor

[VM_fronty]

1 => 1,Prodej

6.2 Soubor `musiconhold.conf`

V tomto souboru se definují sekce, které je potom možno přiřadit frontě, agentovi nebo aplikaci `MusicOnHold()`. Asterisk standardně nepodporuje formát mp3, ale využití formátu mp3 může být dosaženo pomocí externí aplikace např. `mpg123`. V tomto souboru stačí definovat tyto dvě položky

mode=files - definuje, že se soubory budou číst z adresáře v jakémkoliv Asteriskem podporovaném formátu,

directory= - určuje adresář, ze kterého se budou číst hudební soubory.

Následuje příklad souboru `musiconhold.conf`

```
[default]
```

```
mode=files
```

```
directory=/var/lib/asterisk/moh/
```

6.3 Soubor `followme.conf`

Služba "následuj mě" slouží ke spojení s osobou, která využívá více telefonních účtů. Ústředna se pokusí postupně spojit se všemi účty volaného. Prohledávají se jednotlivé klapky, podle pořadí v jakém jsou uvedeny v souboru `followme.conf`. Vybrané možnosti nastavení v sekci `global` jsou uvedeny níže

takecall - klávesa nebo klávesy pro přijetí hovoru volaným,

declinecall - klávesa nebo klávesy pro zamítnutí hovoru volaným.

V definici ID hledaného je možno definovat následující možnosti

music - hudba, která bude přehrávána volajícím, dokud bude ústředna hledat volaného uživatele.

context - definuje kontext ve kterém bude volaný hledán.

number - klapka, která bude v definovaném kontextu vyzkoušena. Druhým parametrem je doba vyzvánění.

Následující příklad je opět použit pro demonstrační číslovací plán.

```
[general]
takecall=>1
declinecall=>2
```

```
[150]
music=>default
context=>hledej
number=>12,20
number=>13,20
number=>14,20
```

6.4 Soubor meetme.conf

V tomto konfiguračním souboru se definují jednotlivé konferenční místnosti (sekce rooms). Syntax jednotlivých místností vypadá takto:

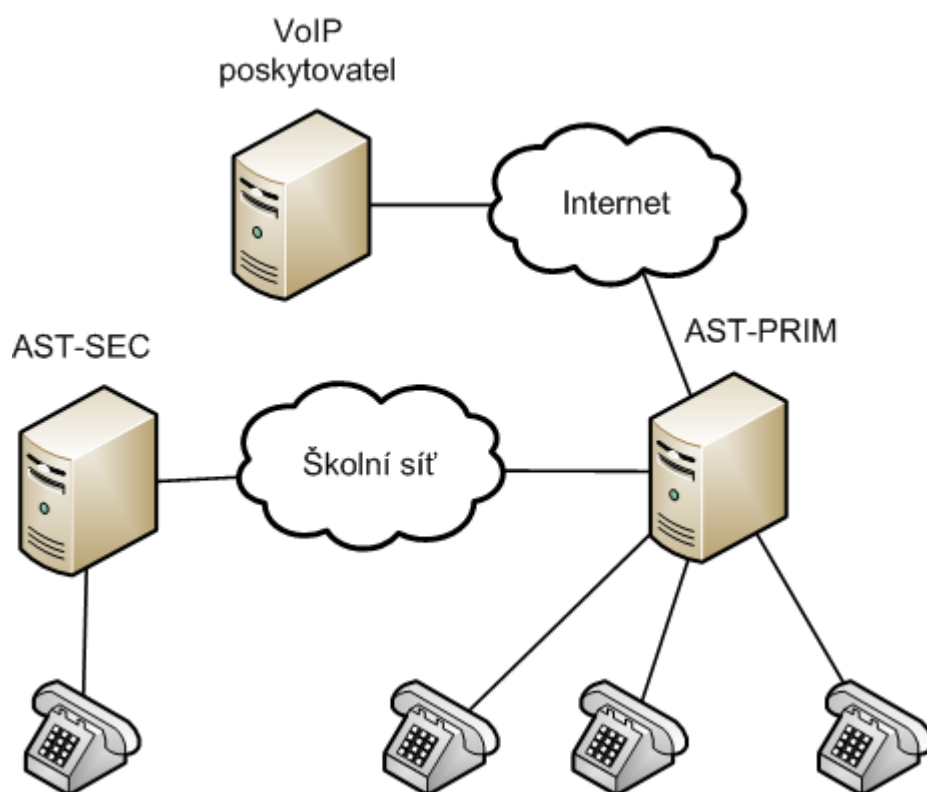
```
conf => confno[,pin][,adminpin].
```

Následuje příklad konfigurace pro využití v demonstračním číslovacím plánu.

```
[rooms]
conf => 100,1
```

7 DEMONSTRAČNÍ ČÍSLOVACÍ PLÁN

Demonstrační číslovací byl testován pomocí dvou ústředen a jednoho SIPového trunku do veřejné sítě. To je vidět obr. 7.1. Konfigurace obou ústředen se nachází na příloženém DVD. Následuje popis pouze číslovacího plánu ústředny ast-prim.



Obr. 7.1: Schema testovacího pracoviště

Tento číslovací plán umožňuje volat mezi jednotlivými vnitřními uživateli. Uživatelé mohou volat do vnější sítě pomocí trunku (trunk_094). Druhým trunkem (ast-sec) mohou volat do druhé ústředny. Příchozí hovory z trunku jsou přesměrovány do kontextu interaktivního hlasového automatu (IVR), pomocí něhož mohou být volající z vnější sítě, přepojeni na konkrétní klapky ústředny. Zatímco hovory z druhé ústředny jsou zpracovány podle volaného čísla. Číslovací plán je rozdělen do několika kontextů, které obsluhují

- [from-internal] - všechna volání od vnitřních uživatelů,
- [ivr-hlavni] - interaktivní hlasový systém,
- [ivr-VM] - podmenu IVR pro hlasovou schránku,
- [C.Prodej] - prodejní linku,

- [C.Pece0Zakazniky] - linku péče o zákazníky,
- [C.TechPodpora] - linku technické podpory,
- [Callback_agent] - klapky pro zpětné volání agentům,
- [from-sip-external] - všechna volání od vnějších uživatelů,
- [from-sip-external-ast-sec] - všechna volání od druhé ústředny,
- [outbound-routes] - volání vnitřních uživatelů mimo ústřednu,
- [hledej] - aplikaci Followme().

Další skupinou kontextů jsou makra

- [macro-dialout-trunk] - pro vytočení zadaného trunku,
- [macro-volani-na-klapku-vm] - pro vytočení vnitřního uživatele a v případě jeho nedosažitelnosti vstoupí do hlasové schránky,
- [macro-systemrecording] - pro záznam kanálu do souboru.

V sekci globals jsou definovány tyto proměnné

```
DELKA_ZVONENI => 10
TRUNK_9 => XXXXXX094
OUT_7 => SIP/ast-sec
OUT_9 => SIP/trunk_094.
```

Význam těchto proměnných je uveden v následující tab. 7.1.

Tab. 7.1: Globální proměnné

Proměnná	Význam
DELKA_ZVONENI	Je použita pro volání na vnitřní klapky
TRUNK_9	Obsahuje číslo trunku 9
OUT_7	SIPový účet trunku 7
OUT_9	SIPový účet trunku 9

7.1 Kontext from-internal

V kontextu `from-internal` jsou definovány všechny klapky, které slouží jednotlivým pobočkám. Velkou skupinou jsou klapky, které spojují volající přímo s jednotlivými vnitřními uživateli. Další klapky provádějí přihlášení agenta, vstup do jednotlivých front, vstup do konference a záznam do souboru atd. Do tohoto kontextu je vložen kontext `outbound-routes`, který umožňuje vnitřním uživatelům volat mimo ústřednu, takto

```
include => outbound-routes.
```

Dále je vložen speciální kontext `parkedcalls`. Ten zajišťuje parkování a přenos hovorů. Konfigurace přenosu a parkování hovorů je provedena v souboru `features.conf` (viz podkapitola 5.3). Pro přenos hovoru je důležité, aby příslušná aplikace `Dial()` byla spuštěna s parametrem `t` nebo `T`. Pokud strana, které byl přenos povolen, stiskne dvakrát `#` Asterisk oznámí "transfer" a čeká, na klapku na kterou má být hovor přenesen v aktuálním kontextu. Jestliže je uživatelem zadána klapka 800, je hovor zaparkován a Asterisk oznámí na kterou klapku hovor zaparkoval. Z této klapky může být vyzvednut libovolným uživatelem, který zadá klapku na které je hovor zaparkován.

Volání mezi vnitřními uživateli se provádí pomocí makra `volani-na-klapku-vm`, které je podrobněji popsáno v podkapitole 7.10.2. Toto makro má jako argument délku zvonění. Jedna z klapek pro volání vnitřnímu uživateli vypadá takto

```
exten => 11,n,Macro(volani-na-klapku-vm,${DELKA_ZVONENI}).
```

Klapka pro vstup do IVR je provedena pomocí aplikace `Goto`:

```
exten => 21,1,Goto(ivr-hlavni,s,1).
```

Následující klapka slouží pro spojení s uživatelem, který se vyskytuje na více telefonních přístrojích. Toho je docíleno pomocí aplikace `FollowMe()` a konfiguračního `followme.conf` (viz podkapitola 6.3), kde je definován kontext 150.

```
exten => 44,1,FollowMe(150|asn)
; s - přehraje zprávu, že se ústředna pokusí najít uživatele
; a - zaznamená jméno volajícího a přehraje volanému
; n - přehraje zprávu o nedosažitelnosti uživatele
```

Asterisk podporuje provolbu DISA. Klapka pro ukázkou funkce DISA:

```
exten => 80,1,Disa(no-password|from-internal).
```

Prvním argumentem je heslo (případně. no-password) a druhým argumentem je kontext ve kterém se vytočená klapka má hledat.

Dále zde existují dvě klapky pro konference jedna konference je vytvořena staticky v souboru **meetme.conf** (viz podkapitola 6.4) a druhá dynamicky. Klapka pro statickou konferenci vypadá následovně

```
exten => 90,1,MeetMe(100).
```

Zatímco klapka pro dynamickou konferenci má tento tvaru

```
exten => 91,1,MeetMe(2|DpMIx).
; D - dynamické vytvoření místnosti
; p - přiřazení PINu prvním volajícím
; M - přehrává hudbu pokud je zde jediný uživatel
; I - ohlášení odpojení a připojení uživatele do konf.
; x - zavři konf., když se poslední uživatel odhlasí
```

Časově podmíněný skok je možno provést pomocí aplikace **GotoIfTime**. Ten může být využit pro sledování pracovní doby následujícím způsobem.

```
exten => 99,1,Answer()
exten => 99,n,Playback(welcome)
exten => 99,n,GotoIfTime(09:00-17:00|mon-fri|*|*?
                    from-internal,99(otevreno),1)
```

```
; zde je zavřeno a zavěsí se
```

```
exten => 99,n,Playback(our-business-hours-are)
exten => 99,n,Playback(goodbye)
exten => 99,n,Hangup()
```

```
; zde je otevřeno může se přepnout do kontextu ivr-hlavni
```

```
exten => 99(otevreno),1,Playback(ha/open)
exten => 99(otevreno),n,Goto(ivr-hlavni,s,1).
```

Pokud je vytočena klapka 99, přehraje ústředna zprávu "welcome", otestuje zda aktuální čas a den v týdnu je uvnitř nastaveného rozsahu. Pokud je, provede se skok na 99(otevreno). Ústředna přehraje "open" a skočí do kontextu ivr-hlavni. V případě, že aktuální čas a den v týdnu neodpovídá rozsahu, přehrají se zprávy "our-business-hours-are" a "goodbye" a ústředna zavěsí. V tab. 7.2 je uveden přehled definovaných klapek.

Tab. 7.2: Definované klapky v kontextu `from-internal`

Číslo klapky	Funkce klapky
11	Volání uživatele Drobek
12	Volání uživatele Trevor
13	Volání uživatele Strakoš
14	Volání uživatele Štolf
15	Volání uživatele Julo
16	Volání uživatele Dushanek
17	Volání uživatele Mišák
18	Volání uživatele Toncek
19	Volání uživatele Čert
21	Vstup do hlasového automatu
44	Hledání uživatele
80	DISA
90	Vstup do statické konference
91	Vstup do dynamické konference
99	Test časového skoku

Správa hlasové schránky se provádí pomocí aplikace `VoiceMailMain()`, jejíž parametrem je číslo hlasové schránky a její kontext. Ještě doplněna aplikací `NoCDR()`, která zamezí ukládání účtovacích informací. Klapka pro správu hlasové schránky volajícího má následující tvar

```
exten => *100,1,NoCDR()
exten => *100,n,VoiceMailMain(${CALLERID(num)}@internal).
```

Uživatel si může do `AstDB` uložit informaci o tom, zda si přeje aby jeho účet používal hlasovou schránku či nikoliv. K tomu je využita aplikace `Set()` a klíčové slovo `DB`, které určuje, že informace budou zapsány do `AstDB`. Definice těchto klapkek vypadá takto

```
; povolení hlasové schránky
exten => *21,n,Set(DB(voicemail/${CALLERID(num)})=1)
; zakázání hlasové schránky
exten => *22,n,Set(DB(voicemail/${CALLERID(num)})=0).
```

Výše uvedené klapky jsou v číslovacím plánu ještě doplněny o oznámení provedené operace a aplikací `NoCDR()`.

Pomocí aplikací `AgentLogin()` a `AgentCallbackLogin()` se přihlašují agenti, kteří jsou definováni v souboru `agents.conf`. Definice klapkek pro přihlášení agenta jsou uvedeny v podkapitole 5.2.

Pro uložení a přehrání souboru jsou určeny klapky *80 a *81. Klapka *80 je určena pro uložení hlasového toku pomocí makra `systemrecording`, které je popsáno v podkapitole 7.10.3. Zatímco klapka *81 tento soubor vždy pouze přehraje (pomocí aplikace `Playback()`). V následující tab. 7.3 je uveden přehled služeb pro vnitřní uživatele.

Tab. 7.3: Služby v kontextu `from-internal`

Číslo klapky	Funkce klapky
*100	Správa hlasové schránky volajícího
*101	Správa libovolné hlasové schránky
*105	Správa hlasových schránek front
*20	Zjištění povolení/zakázání hlasové schránky
*21	Povolení hlasové schránky
*22	Zakázání hlasové schránky
*65	Přihlášení agenta
*66	Přihlášení agenta pro zpětné volání
*67	Odhlášení agenta pro zpětné volání
*80	Nahrávání do souboru
*81	Přehrání uloženého souboru

7.2 Kontext `ivr-hlavni`

Tento kontext obsahuje interaktivní hlasový automat, který přehraje volajícímu menu a volající se rozhodne, kterou službu požaduje a podle toho volající stiskne číslo své volby a ústředna jej přepojí. Jednotlivé volby jsou uvedeny v následující tab. 7.4.

Na počátku kontextu se nastaví proměnná `TIMEOUT(digit)`, což je čas v sekundách, po jakém bude Asterisk považovat volbu za kompletní. Pokračuje se nastavením proměnné `LOOPCOUNT=0`, spuštěním aplikace `Background()` se začne přehrávat menu. Nyní se čeká na uživatelskou volbu `WaitExten()`.

Pokud volající neprovede volbu a byl překročen nastavený časový limit (argument aplikace `WaitExten()`), skočí se na speciální klapku `t` a zde se přehraje hláška "time is up". Následuje skok na klapku `loop`, kde se zvýší hodnota proměnné `LOOPCOUNT` o jedničku. Pokud je hodnota stále nižší než 2, menu se opakuje (skok na prioritu `n(loop)`), ale pokud je hodnota vyšší než 1 přehraje se zpráva "goodbye" (skok na prioritu `hang`) a bude zavěšeno. Ještě je zde speciální klapka `i`, na kterou se skočí pokud volající provede neplatnou volbu, což je uživateli sděleno hláškou "invalid" a

Tab. 7.4: Číselné volby definované v `ivr-hlavni`

volba volajícího	funkce klapky
1	Prodejní linka
2	Centrum péče o zákazníky
3	Linka technické podpory
7	Vstup do konference 100
8	Podmenu pro ovládání hlasové schránky
9	DISA
11	Volání uživatele Drobek
12	Volání uživatele Trevor
13	Volání uživatele Strakoš

pokračuje se skokem na klapku `loop`. Pokud se menu již dvakrát opakovalo ústředna přehraje hlášku "goodbye" a zavěsí (skok na prioritu `hang`).

7.3 Kontext `ivr-VM`

Tento kontext je podmenu hlavního IVR. Volby tohoto menu jsou uvedeny v následující tab. 7.5.

Tab. 7.5: Číselné volby definované v `ivr-VM`

Volba volajícího	Funkce klapky
0	Zjištění povolení/zakázání hlasové schránky
1	Povolení hlasové schránky
2	Zakázání hlasové schránky
#	Skok do hlavního menu

IVR v tab. 7.5 funguje v podstatě stejně jako `ivr-hlavni` s tím rozdílem, že po skončení smyčky se nezavěsí, ale skočí do `ivr-hlavni`.

7.4 Kontexty pro fronty

Jsou definovány tři kontexty pro obsluhu front s názvy `C_Prodej`, `C_Pece0Zakazniky` a `C_TechPodpora`. Tyto kontexty fungují stejně, liší se pouze v informacích, které jsou přehrávány volajícímu a frontou do které je volající přiřazen.

Nejprve volající obdrží informaci o tom, že vstoupil do fronty ("in the queue"). Aplikací `Queue()` je do příslušné fronty přiřazen. Pokud ve frontě není obslužen, je volající pomocí aplikace `VoiceMail()` přeměrován do hlasové schránky příslušné fronty. Tyto fronty mají v souboru `voicemail.conf` definován vlastní kontext `VM_fronty` (viz podkapitola 6.1). Následuje příklad kontextu pro obslužení prodejní linky.

```
[C_Prodej]
exten => s,1,Playback(welcome)
exten => s,n,Playback(in-the-queue)
exten => s,n,SayDigits(1)
exten => s,n,Queue(Prodej|t) ; t - umožňuje přenos hovoru volaným
exten => s,n,VoiceMail(1@VM_fronty)
exten => s,n,Playback(vm-goodbye)
exten => s,n,Hangup()
```

7.5 Kontext Callback-agents

Tento kontext je využíván aplikací `AgentCallbackLogin()`. Když se aplikace dotáže přihlašovaného na klapku, na které bude zastizitelný, musí volající uvést klapku, která je v tomto kontextu. Jsou zde uvedeny klapky na všechny vnitřní uživatele.

7.6 Kontext from-sip-external

V kontextu `from-trunk` je definováno kam se mají přepojit hovory, které přicházejí z trunku od poskytovatele. Hovory jsou přeměrovány do IVR. Kontext končí vzorem čísel "_.", který zachytí všechna ostatní volání a přehraje volajícímu zprávu, která mu oznámí, že volané číslo není obsluhováno. V demonstračním číslovacím plánu vypadá tento kontext následovně

```
; tyto hovory jdou do ivr
exten => ${TRUNK_2},1,Answer()
exten => ${TRUNK_2},n,GoTo(ivr-hlavni,s,1)

exten => _.,1,Playback(ss-noservice).
```

7.7 Kontext `from-sip-external-ast-sec`

Kontext slouží příchozím volání z ústředny `ast-sec`. Obsahuje dvě klapky. Klapka 200, která přehraje hlášku "welcome" a číslo "1" a klapku 555, která příchozí hovor přesměruje do kontextu `ivr-hlavni`.

7.8 Kontext `outbound-routes`

Kontext `outbound-routes` slouží pro volání vnitřních uživatelů mimo ústřednu (outbound). Tito uživatelé mohou volat s prefixem **7** nebo **9**. Zavolá se makro `dialout-trunk` s parametry prefix, volané číslo a parametry aplikace `Dial()`. Makro je podrobněji popsáno v podkapitole 7.10.1. Volané číslo se vytvoří tak, že se odsekne prefix pomocí `":1"`. Volání makra vypadá následovně

```
exten => _7.,1,Macro(dialout-trunk,7,${EXTEN:1},30)
exten => _9.,1,Macro(dialout-trunk,9,${EXTEN:1},30).
```

7.9 Kontext `hledej`

Kontext `hledej` je využíván aplikací `FollowMe()`. Jsou v něm uvedeny klapky na kterých se má uživatel hledat. Tyto klapky odpovídají klapkám uvedeným v souboru `followme.conf` (viz podkapitola 6.3).

7.10 Makra

7.10.1 `macro-dialout-trunk`

Makro `macro-dialout-trunk` má tři argumenty. První argument (**ARG1**) určuje který trunk bude vytočen, druhým argumentem (**ARG2**) je volané číslo a posledním argumentem (**ARG3**) jsou volby se kterými bude trunk vytočen. Tyto argumenty jsou v tomto pořadí přiřazeny proměnným `$DIAL_TRUNK`, `$DIAL_NUMBER` a `$DIAL_TRUNK_OPTIONS`. Přiřazení je provedeno kvůli přehlednosti. Následuje přiřazení čísla pobočky, ze které je voláno, do proměnné `CDR(accountcode)`, která se používá pro účtování hovorů. Těsně před vytočením trunku se provede vynulování `CDR` záznamů (aplikace `NoCDR()`) a poté následuje vytočení trunku. V sekci `globals` musí být definována proměnná `OUT_1`, která určuje technologii a jméno trunku tak, jak je definováno v souboru `sip.conf` (např. `OUT_1 => SIP/trunk_094`). Vytočení trunku má tvar:

```

exten => s,1,Set(DIAL_TRUNK=${ARG1}) ; číslo trunku 1
exten => s,n,Set(DIAL_NUMBER=${ARG2}) ; volané číslo
exten => s,n,Set(DIAL_TRUNK_OPTIONS=${ARG3}) ; parametry dial()

exten => s,n,Set(CDR(accountcode)=${CALLERID(num)})
exten => s,n,ResetCDR() ; snuluje CDR
exten => s,n,Dial(${OUT_${DIAL_TRUNK}}/${DIAL_NUMBER},
                ${DIAL_TRUNK_OPTIONS})
exten => s,n,Goto(s-${DIALSTATUS},1)

```

Následuje aplikace `GoTo()`, která provede skok podle stavu kanálu jakým vytáčení skončilo. Dále jsou ošetřeny jednotlivé stavy kanálu (`s-BUSY`, `s-NOANSWER`, `s-CANCEL`).

7.10.2 macro-volani-na-klapku-vm

Makro `volani-na-klapku-vm` vytočí SIPový účet, který je předán pomocí proměnné `MACRO_EXTEN` (klapka ze které bylo makro spuštěno) s parametrem `ARG1`, který představuje délku zvonění.

Pokud volaný uživatel neodpoví nebo není vůbec dostupný otestuje zda má volaný povolenou hlasovou schránku. Když je hlasová schránka povolena, je spuštěna aplikace `Voicemail()`, která vstoupí do hlasové schránky volaného. V opačném případě ústředna pouze přehraje hlášku "goodbye". Struktura makra je vidět níže.

```

exten => s,1,Dial(SIP/${MACRO_EXTEN},${ARG1},rtT)
exten => s,n,Goto(s-${DIALSTATUS},1)

exten => s-NOANSWER,1,GotoIf($[${DB(voicemail/${MACRO_EXTEN})} = 1]?
                        vm_yes:vm_no)
exten => s-NOANSWER,n(vm_yes),Voicemail(${MACRO_EXTEN}@internal|u)
exten => s-NOANSWER,n(vm_no),NoCDR()
exten => s-NOANSWER,n,Playback(vm-goodbye)
exten => s-NOANSWER,n,Hangup()

exten => s-BUSY,1,GotoIf($[${DB(voicemail/${MACRO_EXTEN})} = 1]?
                        vm_yes:vm_no)
exten => s-BUSY,n(vm_yes),Voicemail(${MACRO_EXTEN}@internal|b)
exten => s-BUSY,n(vm_no),NoCDR()
exten => s-BUSY,n,Playback(vm-goodbye)
exten => s-BUSY,n,Hangup()

```

```
exten => _s.,1,Goto(s-NOANSWER,1)
```

Pokud se spojení nepodařilo, provede se skok (aplikace `Goto(s-$DIALSTATUS)`) na klapku `s-NOANSWER`, `s-BUSY` anebo `_s.`, podle toho jakým stavem skončilo vytáčení. Poté se otestuje zda má uživatel povolenou hlasovou schránku (pomocí aplikace `GotoIf()` a dotazu `DB(voicemail/$MACRO_EXTEN)`). Když je hlasová schránka povolena vstoupí se do ní pomocí aplikace `Voicemail()`. V opačném případě se ústředna rozloučí hláškou "goodbye". Stav `NOANSWER` a `BUSY` jsou rozlišeny pouze proto aby ústředna sdělila uživateli, že volaný je nedostupný ("unavailable") anebo že je za neprázdněn ("busy").

7.10.3 macro-systemrecording

Makro `macro-systemrecording` slouží k uložení audia z kanálu do souboru, jehož název se skládá z čísla volajícího a řetězce "-recording" a bude uložen v adresáři `/tmp/`. Uložení tohoto souboru se provádí pomocí aplikace `Record()`. Toto makro se skládá, ze tří částí `dorecord`, `docheck` a `confmenu`. Při volání makra argument (`ARG1`) udává, která z částí má být provedena jako první. V části `confmenu` má uživatel tři volby "*" pro záznam audia z kanálu, "1" pro přehrání zaznamenaného souboru a "2" pro zavěšení.

8 ÚLOHA 1 - SIP

8.1 Připojení k Asterisku

K asterisku je možno se připojit pomocí protokolu SSH. Používá se aplikace Putty, která implementuje protokol SSH. Po spuštění aplikace stačí vyplnit IP adresu případně doménové jméno, zvolit protokol SSH a kliknout Open. Poté se objeví výzva vzdáleného systému k zadání přihlašovacích údajů, pokud jsou údaje v pořádku objeví se shell vzdáleného systému.

8.2 Asterisk CLI

Asterisk CLI (Command Line Interface) umožňuje správci pracovat s běžícím Asteriskem. Je to velice užitečné rozhraní pro diagnostiku ústředny. CLI spustíme příkazem `asterisk -rvvv`, přičemž volba `'r'` určuje připojení k běžícímu Asterisku a volba `"v"` znamená úroveň "upovídání" (čím více `"v"` je přidáno, tím více informací je poskytováno). Poté se objeví prompt `HOSTNAME*CLI>`, který indikuje konzoli Asterisku.

8.3 Konfigurace softwarového telefonu X-Lite

Po spuštění telefonu je nejdůležitější nakonfigurovat v telefonu SIP účet, kterým se telefon zaregistruje na ústředně. Nastavení SIP účtů lze nalézt, pokud se pravým tlačítkem myši klikne kdekoli na telefon (mimo jakékoliv tlačítko). Objeví se pop-up menu, kde se vybere volba "SIP Account Settings...". Pokud nějaký účet existuje odebereme jej pomocí tlačítka "remove".

Prvním krokem pro přidání SIP účtu je tlačítko "Add...". Objeví se vlastnosti účtu. Vyplníme následující položky

- Display Name - jméno, které se bude zobrazovat volanému,
- User name - přihlašovací jméno, které je definováno na ústředně,
- Password - heslo pro účet,
- Domain - IP adresa nebo doménové jméno ústředny.

Je vhodné vypnout kontrolu hlasové schránky. To se provádí na záložce Voicemail odškrtnutím "Check for voicemail". Vše potvrdíme. X-Lite oznámí, že se registroval u ústředny.

8.4 Postup

1. Připojte se pomocí Putty k Asterisku. Pro přihlášení jsou zapotřebí následující údaje:
 - user: ,
 - password: ,
 - IP: .
2. Pomocí Putty se podruhé připojte k Asterisku. V terminálu spusťte Asterisk CLI s verbose level 3, ve kterém budete sledovat dění na ústředně, případně ústřednu restartovat pomocí příkazu `restart now` po každé změně konfigurace.
3. Vytvořte v souboru **sip.conf** jednoho uživatele jehož jméno bude 100 a povolte mu kodeky gsm a alaw.
4. Vyberte si telefon a nakonfigurujte jeho přihlášení k Asterisku. Poté telefon oznámí, že se přihlásil a Asterisk CLI oznámí tuto registraci též.
5. Vytvořte v souboru **extensions.conf** klapku 555, která bude uživateli opakovat jeho slova, k tomu slouží aplikace `Echo()`. Důležité je, aby jméno vašeho kontextu, který vytvoříte odpovídalo jménu kontextu, který jste definovali v souboru **sip.conf**.
6. Vyzkoušejte funkci, vytvořte vámi definovanou klapku, přitom sledujte co se děje v CLI. Aplikaci `Echo` ukončíte stiskem klávesy ”#” na telefonním přístroji.
7. Vytvořte další uživatele (101, 102, atd.) a nakonfigurujte další zařízení pro přihlášení k Asterisku.
8. Rozšiřte číslovací plán o další klapky na kterých bude možno zastihnout přihlášené uživatele (101 pro prvního uživatele, 102 pro druhého uživatele atd.).
9. Ověřte funkci tj. všichni uživatelé si mohou vzájemně volat. Při testování sledujte co se děje v Asterisk CLI.

8.5 Zachycení SIP zpráv

Wireshark

Výměnu zpráv mezi Asteriskem a SIPovým telefonem lze zkontrolovat pomocí aplikace Wireshark (<http://www.wireshark.org>), což je paketový sniffer a analyzátor

síťových protokolů. Používá se při ladění problémů v počítačových sítích, vývoji aplikací, studium síťových protokolů atd.

1. Nejprve vypněte softwarový telefon X-lite.
2. Na klientském počítači spusťte aplikaci Wireshark. V hlavním menu vyberte Capture -> Interfaces. Objeví se okno s jednotlivými síťovými zařízeními. U zařízení ze kterého si přejete stiskněte tlačítko Options. Do řádku Capture Filter napište port 5060. Poté stiskněte tlačítko Start.
3. Nyní spusťte softwarový telefon X-lite. V packet listu se objeví několik paketů, které představují SIP registraci softwarového telefonu. Následně zachytávání paketů zastavte (Capture -> Stop). Prohlédněte si jednotlivé pakety. Ve výpisu se objeví i SIPové zprávy SUBSCRIBE, pokud jste v telefonu nevypnuli kontrolu hlasové schránky.
4. Stejným způsobem jakým jste zachytili registraci zachyťte i volání z nebo na softwarový telefon X-lite. Zachycené pakety opět zanalyzujte.

9 ÚLOHA 2 - IVR

9.1 Interactive voice response

Interaktivní hlasová odezva je technologie, která umožňuje volajícímu provádět akce na ústředně pomocí DTMF signálů. IVR detekuje stisknuté klávesy a na základě této volby provádí definované akce (např. vstup do hlasové schránky, vytočit uživatele atd.). Ústředna komunikuje s volajícím pomocí připravených nebo dynamicky vygenerovaných (pomocí systému TextToSpeech) hlášek, podle kterých se uživatel v systému orientuje. IVR má široké využití v praxi např. telefonní bankovníctví, hlasování přes telefon, automatický průvodce (Automated attendant).

IVR se v Asterisku vytváří jako jakýkoliv jiný kontext v číslovacím plánu. Při vstoupení do kontextu IVR se přehraje uvítací hláška a následně hláška obsahující možnosti uživatele, pomocí aplikace `Background`, která je přímo schopna přijímat DTMF signály stisknutých kláves volajícího během přehrávání, což aplikace `Playback` neumí. Všechny výše uvedené aplikace by měly běžet na standardní klapce `s`. Pokud volající provede volbu, ústředna spustí aplikaci, která je definována na zvoleném čísle. Pro ošetření vyjimečných stavů se používají dvě standardní klapky `t` a `i`. Funkce speciálních klapek je následující

`s` - (start) - na tuto klapku se skočí pokud kontext právě začal,

`i` - (invalid) - na tuto klapku se skočí pokud volající zvolí nedefinovanou klapku,

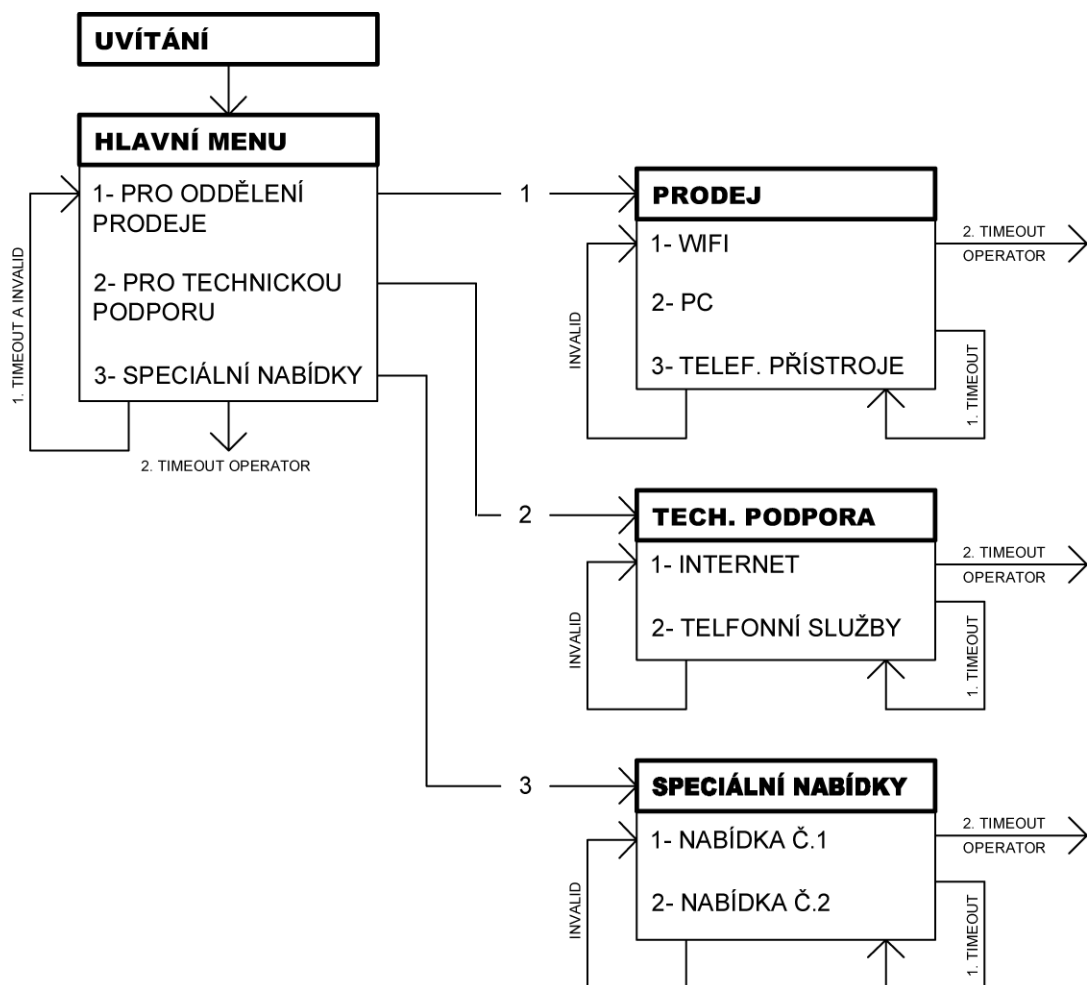
`t` - (timeout) - na tuto klapku se skočí pokud je překročen určený čas.

9.2 Postup

1. Připojte se pomocí Putty k Asterisku. Pro přihlášení jsou zapotřebí následující údaje:
 - user: ,
 - password: ,
 - IP: .
2. Pomocí Putty se podruhé připojte k Asterisku. V terminálu spusťte Asterisk CLI s verbose level 3, ve které budete sledovat dění na ústředně. Případně ústřednu restartovat pomocí příkazu `restart now` po každé změně konfigurace.

3. Vytvořte v souboru **sip.conf** jednoho uživatele (např. 100) a povolte mu kodeky gsm a alaw.
4. Vyberte si telefon a nakonfigurujte jeho přihlášení k Asterisku. Poté telefon oznámí, že se přihlásil. Asterisk CLI oznámí tuto registraci též.
5. Vytvořte v souboru **extensions.conf** kontext IVR-hlavni, který přehraje hlášku **welcome** (pomocí aplikace `Playback()`).
6. Do tohoto kontextu skočte z vámi definované klapky (např. 21) z kontextu definovaném v **sip.conf** (pomocí aplikace `Goto()`). Ověřte funkčnost.
7. Na následující prioritě nastavte proměnnou `POCET_OPAKOVANI` na hodnotu 0 (pomocí aplikace `Set()`). Této prioritě dejte nějaké jméno (např. begin). Dále přehrajte uživateli hlášku "menu" (pomocí aplikace `Background()`). Tuto prioritu si též pojmenujte (např. menu). Po ukončení přehrávání je třeba počkat určitou dobu na volbu uživatele toho docílíte pomocí aplikace `WaitExten()`. Všechny předcházejí aplikace by se měli nacházet na speciální klapce **s**.
8. Nyní vytvořte jednotlivé volby (klapky) hlavního menu podle obr. 9.1, které může uživatel provést. Na těch to klapkách vytvořte skoky do tří dílčích kontextů (pomocí aplikace `GoTo()`). Tedy každá dílčí sekce bude tvořena jedním kontextem. Jména mohou být např. **Prodej**, **TechPod** a **Nabidky**. Identifikaci kde se volající nachází můžete provést pomocí `Playback(číslo)`.
9. Vytvořte standardní klapku **t**. Na této klapce inkrementujte hodnotu proměnné `POCET_OPAKOVANI`. Následovat bude test proměnné `POCET_OPAKOVANI` zda je menší než 1. Pokud ano ústředna skočí na přehrávání menu (pomocí `GotoIf()` a pojmenované priority), jinak přepojí volajícího přepojí na operátora. Operátora vytvořte v souboru **sip.conf** (libovolný účet).
10. V kontextu IVR vytvořte standardní klapku **i** (s prioritou 1), která bude fungovat stejně jako klapka **t**.
11. Tímto byste měli mít hotové hlavní menu. Nyní stejným způsobem vytvořte zbývající tři kontexty podle obr. 9.1. Poté IVR otestujte a přitom sledujte co děje v CLI. Informace z CLI mohou být použity pro odladění pokud vám přece jen něco nefunguje.

Struktura hlasového automatu je vidět na obr. 9.1.



Obr. 9.1: IVR menu pro call centrum

Níže je uvedena jednoduchá struktura IVR.

```

[ivr-hlavni]
; začíná smyčka menu
exten => s,1,Answer()
exten => s,n(begin),Set(LOOPCOUNT=0)
; přehraj možnosti menu
exten => s,n(loop),Background(custom/menu)
exten => s,n,WaitExten(5)

; volby uživatele
exten => 1,1,Goto(nejaky_kontext_a,1,1)
exten => 2,1,Goto(nejaky_kontext_b,1,1)

; byl překročen časový limit
  
```

```
exten => t,1,Goto(loop,1)
```

```
; byla provedena špatná volba
```

```
exten => i,1,Goto(loop,1)
```

```
; skoč na začátek pokud LOOPCOUNT > x
```

```
exten => loop,1,Set(LOOPCOUNT=${${LOOPCOUNT} + 1})
```

```
exten => loop,n,GotoIf(${${LOOPCOUNT} > 1}?operator,1:s,loop)
```

```
; přepoj na operátora
```

```
exten => operator,1,Goto(kontext_operatora,1,1)
```

10 ÚLOHA 3 - CALL CENTRUM

10.1 Postup

1. Připojte se pomocí Putty k Asterisku. Pro přihlášení jsou zapotřebí následující údaje:
 - user: ,
 - password: ,
 - IP: .
2. Vytvořte v souboru `sip.conf` jednoho uživatele a povolte mu kodeky `gsm` a `alaw`.
3. Pomocí Putty se podruhé připojte k Asterisku. V terminálu spusťte Asterisk CLI s verbose level 3, ve které budete sledovat dění na ústředně. Případně ústřednu restartovat pomocí příkazu `restart now` po každé změně konfigurace.
4. Vyberte si telefon a nakonfigurujte jeho přihlášení k Asterisku. Na ústředně jsou nakonfigurovány účty 11 až 19 s hesly "asdf". Poté telefon oznámí, že se přihlásil. Asterisk CLI oznámí tuto registraci též.
5. V souboru **agents.conf** v sekci **agents** přiřadte hudbu na pozadí (pomocí parametrů `musiconhold`). Tato hudba na pozadí je definována v souboru **musiconhold.conf**. Potom vytvořte skupinu, s číslem 1, dvou agentů (511 a 512). Přiřadte jim hesla (pro jednoduchost raději stejná).
6. Nyní je nutné v číslovacím plánu (**extensions.conf**) vytvořit klapku *65, která umožní agentovi se přihlásit (pomocí aplikace `AgentLogin()`). Tuto klapku vytvořte v kontextu `internal`, který definovaným uživatelům přiřazen v souboru **sip.conf**. Ověřte funkčnost (vytočte klapku *65). Po zadání přihlašovacích údajů bude oznámeno úspěšné přihlášení a začne hrát hudba na pozadí.
7. V dalším kroku vytvořte v souboru **queues.conf** frontu `Prodej`. Nastavte jí hudbu na pozadí (`musiconhold`) a jako člena uveďte skupinu agentů číslo 1 (`member => Agent/@1`). Od této chvíle budou tito agenti přijímat hovory z právě definované fronty.
8. Opět se vraťte k číslovacímu plánu a vytvořte klapku 51 na které bude volající uživatel přiřazen do fronty `Prodej` pomocí aplikace `Queue()`. Ověřte

funkčnost. Přihlaste agenta a potom vytočte klapku 51. Hovor bude spojen s agentem, který je přihlášen k dané frontě.

9. Vytvořte další dvě fronty a čtyři agenty podle následující tabulky tab. 10.1. Ještě vytvořte klapku 52 pro dosažení fronty **PeceOZakazniky** a klapku 53 pro zařazení volajícího do fronty **TechPodpora**.
10. Nakonec vytvořte v číslovacím plánu (**extensions.conf**) kontext IVR, podle vzoru uvedeného níže.

```
[ivr]
exten => s,1,Answer()
exten => s,n,Background(custom/menu)
exten => s,n,WaitExten(5)

exten => 1,1,Goto(kontext,51,1)
exten => 2,1,Goto(kontext,52,1)
exten => 3,1,Goto(kontext,53,1)
```

Tab. 10.1: Tabulka front a agentů

Skupina	Agenti	Fronta kterou obsluhují
1	511, 512	Prodej
2	521, 522	PeceOZakazniky
3	531, 532	TechPodpora

11 ZÁVĚR

V diplomové práci bylo nejprve obecně pojednáno o ústředně Asterisk. Bylo popsáno jaké podporuje signalizační protokoly, kodeky a možnosti jejího nasazení. Následuje popis signalizačního protokolu SIP, který je v současné době hojně využíván.

V diplomové práci byl vytvořen číslovací plán, který demonstruje jakým způsobem se číslovací plány konfiguruje. Je v něm ukázáno jakým způsobem se hovory směřují z ústředny a do ústředny. Mezi služby vytvořeného číslovacího plánu patří parkování a předávání hovorů, hlasová schránka, konference, interaktivní hlasový automat a fronta hovoru.

Výsledkem této diplomové práce jsou tři laboratorní úlohy. První úloha se zabývá konfigurací SIPových klientů a jejich vzájemnému spojování. Druhá úloha si klade za cíl osvětlit, jak je v Asterisku možné vytvořit interaktivní hlasový automat. Poslední úloha se zabývá vytvořením call centra pomocí Asterisku.

Ústřednu Asterisk lze vřele doporučit pro nasazení v malých a středních firmách. Pořizovací cena je oproti hardwarovým ústřednám velice nízká. Asterisk dokáže plně nahradit stávající pobočkové ústředny. Při nasazování Asterisku je třeba mít na paměti, že síť by měla zajišťovat QoS.

LITERATURA

- [1] VAN MEGGELEN, J., MADSEN, L., SMITH, J. *AsteriskTM: The Future of Telephony, Second Edition*. USA: O'Reilly Media, Inc., 2007. ISBN-13: 978-0-596-51048-0
- [2] *[asterisk]/trunk/configs* [online]. [cit. květen 2008].
<<http://svn.digium.com/view/asterisk/trunk/configs/>>
- [3] Digium, Inc. *About Asterisk* [online]. c2008, [cit. květen 2008].
<<http://www.asterisk.org/support/about>>
- [4] SINNREICH, H., JOHNSTON, A. B. *Internet Communications Using SIP: Delivering VoIP and Multimedia Services with Session Initiation Protocol*. Canada: Wiley, 2006. ISBN 978-0471776574
- [5] Telecom Matters Corp. *Call Center and ACD (Automatic Call Distribution)* [online]. c2006, [cit. květen 2008]. <<http://www.telecommatters.net/node/10>>
- [6] IETF *RFC 3261 - SIP: Session Initiation Protocol* [online]. [cit. květen 2008].
<<http://www.ietf.org/rfc/rfc3261.txt>>
- [7] IETF *RFC 4566 - SDP: Session Description Protocol* [online]. [cit. květen 2008]. <<http://www.ietf.org/rfc/rfc4566.txt>>