

Asterisk a jeho použití

Technická zpráva

Issue date: 30.10.2005
Authors: Tomáš Wija, David Zukał, Miroslav Vozňák

Cesnet, z.s.p.o.
Zikova 4
160 00 Praha

CESNET © 2005

Obsah:

- 1 Softwarové ústředna Asterisk
 - 1.1 Co je Asterisk
 - 1.2 Požadavky na OS Linux
 - 1.3 Podporované technologie
 - 1.3.1 Zaptel pseudo TDM rozhraní
 - 1.3.2 Non-Zaptel rozhraní
 - 1.3.3 Paket voice protokoly
 - 1.4 Podporované kodeky
- 2 Architektura
- 3 Dialplan – číslovací plán
 - 3.1 Organizace dialplan
 - 3.1.1 [general]
 - 3.1.2 [globals]
 - 3.1.3 Používání proměnných
 - 3.1.4 Proměnné a výrazy
 - 3.1.5 Funkce manipulace s řetězci
 - 3.2 Context a extensions
 - 3.2.1 Používání context
 - 3.2.2 Začlenění context
 - 3.3 Extensions
 - 3.3.1 Jména extensions
 - 3.3.2 Předdefinovaná jména extension
 - 3.3.3 Definování extension
 - 3.3.4 Vzory extension
 - 3.4 Pomocné funkce
 - 3.4.1 Znovu zavedení
 - 3.4.2 Problematika rozdělení souborů
 - 3.4.3 Forwarding na jiný Asterisk
 - 3.4.4 Ovládání extensions.conf zvenku
 - 3.4.5 Použití makra pro vytvoření extension
 - 3.5 Předdefinované kanálové proměnné
 - 3.6 Aplikačně specifické proměnné
 - 3.7 Proměnné specifické pro makra

- 3.8 Proměnné prostředí unix
- 4 Kanály Asterisku
 - 4.1 Co to je „kanál“?
 - 4.2 Typy kanálů
 - 4.2.1 Kanál Agent
 - 4.2.2 Kanál H.323
 - 4.2.3 Kanál IAX
 - 4.2.4 Kanál Local
 - 4.2.5 Kanály Modem
 - 4.2.6 Kanálový modul SIP
 - 4.2.7 Kanál vpb
 - 4.2.8 Zap kanálový modul
- 5 Instalace Asterisku
- 6 Konfigurační soubory
 - 6.1 Dialplan (extensions.conf)
 - 6.2 Session Initiation Protocol (sip.conf)
 - 6.3 Inter-Asterisk Exchange (iax.conf)
- 7 Konfigurace SIP klientů
 - 7.1 optiPoint 400 Standard – HW
 - 7.2 Budge Tone – 100 - HW
 - 7.3 SJ Phone - SW
 - 7.4 Windows Messenger – SW
 - 7.5 X-lite - SW
- 8 Konfigurace IAX klientů
 - 8.1 Ethernet Phone AT-320
 - 8.2 DIAX - SW
 - 8.3 FireFly - SW
- 9. Odkazy na Internet
 - 9.1 SW klienti

1. Softwarová ústředna ASTERISK

1.1 Co je Asterisk

Jestliže se zajímáte o VoIP, nemáte hrůzu v očích při vyslovení „open source řešení“ a hodláte ušetřit několik desítek tisíc, je Asterisk určen právě vám. Oficiálně je Asterisk open source hybrid TDM a packet voice PBX, jedná se o IVR (Interactive Voice Response) platformu s funkcí Automatic Call Distribution (ACD). Neoficiálně jde možná o jedno z „nejsilnějších“, flexibilních a rozšiřitelných řešení v oblasti integrovaného telekomunikačního softwaru. Jde tedy o kompletní open source softwarovou PBX, běžící na platformách Linux a Unix, poskytující veškeré vlastnosti, které byste očekávali od PBX. Jedná se o obecnou distribuci pod podmínkami GNU (General Public Licence). Povolenou výjimku tvoří spojení s OpenH323 projektem a to za účelem dostupnosti H.323 podpory. Systém je navržen tak, aby vytvořil rozhraní telefonnímu hardwaru, softwaru a libovolné telefonní aplikaci.

Asterisk může být mimo jiné použit v těchto aplikacích:

- Různorodá VoIP gateway (MGCP, SIP, IAX, H.323)
- Pobočková ústředna (PBX)
- Voicemail služby s adresářem
- Interaktivní hlasový průvodce (IVR) server
- Softwarová ústředna (Softswitch)
- Konferenční server
- Packet voice server
- Šifrování telefonních nebo faxových volání
- Překlad čísel
- Aplikace Calling card
- Prediktivní volič (Predictive dialer)
- Řazení volání do front se vzdáleným zprostředkovatelem
- Vzdálené „kanceláře“ pro existující PBX

1.2 Požadavky na OS Linux

Asterisk je známý tím, že běží na mnoha platformách OS, nicméně hlavní vývojářskou platformou je Linux. Jestliže bude Asterisk používán pouze pro VoIP, nebo jste-li příliš pohodlní používat externí media gateway určené pro připojení konvenčních telefonních zařízení, pak můžete vybírat z většího množství systémů (například FreeBSD, Mac OS X a Solaris).

Systém je navržen tak, aby pracoval na linuxovském jádře verze 2.4, nicméně je zde i nějaká podpora pro jádro 2.6. S jádrem verze 2.4 pracují například RedHat (7.x, 8, 9 a Enterprise), Debian (i386 a PPC architektura), Fedora, Gentoo, Suse (verze 8 a 9), Mandrake (9.0) a Slackware (9.0, 9.1 a 10.0). Pokoušíte-li se vybudovat stabilní systém, je doporučeno použití jádra 2.4.

1.3 Podporované technologie

Systém je navržen tak, aby povoloval použití nových rozhraní a umožňoval snadno přidávat nové technologie. Jeho cílem je podpora veškerých možných typů současných i budoucích telefonních technologií. Obecně jsou rozhraní rozdělená do tří základních skupin:

- Zaptel hardware
- non-Zaptel hardware
- packet voice.

1.3.1 Zaptel pseudo TDM rozhraní

Tvorba nenákladných rozhraní nebyla vůbec jednoduchým úkolem. Tradiční TDM hardware (např. Dialogic, později majetkem Intelu) byl patentován a také byl příliš drahý. Pro dosažení vymezeného cíle bylo přistoupeno ke zcela nové myšlence. Místo, aby zpracování TDM probíhalo hardwarově, byl přidán hostitelský procesor a Asterisk pracoval s tímto procesorem. Jak se CPU stávaly stále rychlejšími a rychlejšími, začalo být rozumnější pro toto TDM zpracování ponechat software využívat hlavní CPU počítače. Po přidání TDM podpory do Asterisku začala firma Zapata Telephony s výrobou pseudo TDM rozhraní, které nazvala Zaptel. Pseudo TDM architektura poskytuje téměř stejnou kvalitu a real-time schopnosti jakou má hardware TDM. Podstatným rozdílem je však podstatně nižší cena a vyšší flexibilita. Zaptel rozhraní dodává firma Digium (<http://www.digium.com>) a to pro různé varianty síťových rozhraní (včetně PSTN, POTS, T1, E1, PRI, PRA, E&M a mnoho dalších).

1.3.2 Non-Zaptel rozhraní

Tato rozhraní poskytnou konektivitu směrem k tradičním telefonním službám, nepodporují však pseudo-TDM komutování.

Zahrnují následující typy rozhraní:

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

1.3.3 Packet voice protokoly

Zatímco rozhraní Zaptel pro spojení serveru přímo k PSTN již pracovalo spolehlivě, bylo potřeba vyřešit komunikaci čistě přes IP (později Frame Relay). Jednalo se o případ, kdy začátek nebo konec hlasové relace vznikl někde jinde. Jedná se o standardní protokoly pro komunikaci přes paketové sítě (IP a Frame Relay) a jsou to jediná rozhraní, která nepožadují specializovaný hardware. Jelikož autor Asterisku (Mark Spencer) neměl v přílišné oblibě patentovaný protokol H.323, rozhodl se navrhnout a realizovat svůj vlastní protokol. Výsledkem je protokol IAX (Inter Asterisk eXchange) jenž se stará o signalizaci a transport packet voice mezi dvěma připojenými uzly. Ačkoliv jméno naznačuje přítomnost Asterisku na obou koncích komunikace, IAX může ve skutečnosti spojit každé dva koncové body podporující tento protokol. Následně byla přidána podpora součinnosti s ostatními VoIP systémy a podpora pro další packet voice protokoly. Jedná se o protokoly SIP, H.323, MGCP

(Media Gateway Control Protocol) a VoFR (Voice over Frame Relay). S takovou podporou se Asterisk stává v současné době ideální PBX na bázi IP pro prostředí s rozmanitými typy používaných telefonů.

1.4 Podporované kodeky

Asterisk podporuje následující kodeky:

- ADPCM
- G.711 μ -law
- G.711 a-law
- G.723.1
- G.726
- G.729
- GSM
- iLBC (internet Low Bitrate Codec)
- LPC10
- Speex

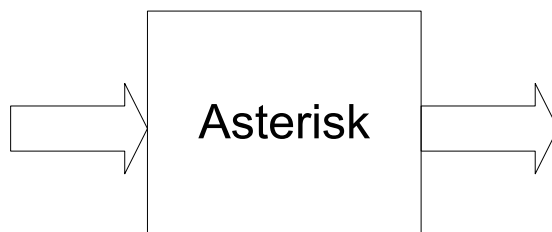
Poznámka: Různí klienti podporují proměnnou periodu vzorkování / paketizaci. **Asterisk** však **podporuje pouze paketizaci 20 ms** v RTP založených protokolech, je tedy klienty potřeba s ohledem na tuto skutečnost nakonfigurovat.

2. Architektura

Architektura systému je v podstatě velmi jednoduchá, ale zcela odlišná od nejčastěji používaných telefonních produktů. Asterisk vystupuje v podstatě jako středový prvek spojující telefonní technologie na jedné straně s telefonními aplikacemi na straně druhé (**Obr. 1**). Vytváří konzistentní prostředí pro rozmístění smíšeného telefonního prostředí.

Telefonní technologie mohou zahrnovat VoIP služby (jako SIP, H.323, IAX a MGCP jak gateway tak telefony), stejně jako tradiční TDM technologie (jako T1, ISDN PRI, analogové POTS a PSTN služby, ISDN BRI, atd.).

Telefonní aplikace zahrnují například přemostění volání, konference, hlasovou poštu, automatickou obsluhu, uživatelské IVR skriptování, parkování hovoru, intercom a další.



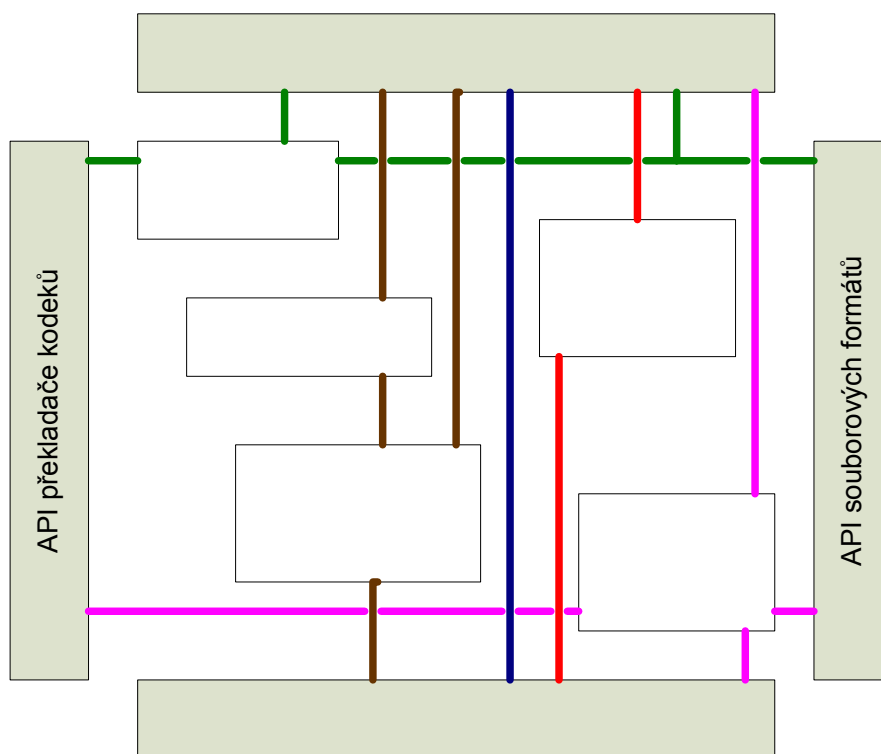
Obr. 1: Obecné umístění Asterisku v systému

Asterisk je pečlivě navržený za účelem poskytnutí maximální flexibility. Okolo systému centrálního jádra PBX jsou definovány specifické API (**Obr. 2**). Toto pokročilé jádro ovládá vnitřní propojení PBX, myšleno specifické protokoly, kodeky a hardwarové rozhraní

telefonních aplikací. To dovolí v Asterisku použít každou vhodnou technologii a hardware (nyní i v budoucnu) za účelem vykonávání základních funkcí - propojování hardwaru a aplikací.

Asterisk jádro vnitřně ovládá tyto položky:

- PBX přepojování (PBX Switching) - podstatou Asterisku je samozřejmě přepojovací systém pobočkové ústředny, spojovací volání mezi různorodými uživateli a automatizovanými úlohami. Přepojovací jádro transparentně spojuje příchozí volání na různých hardwarových a softwarových rozhraních.
- Spouštěč aplikací (Application Launcher) - spouští aplikace zajišťující služby jako jsou například hlasová pošta, přehrání souboru a výpis adresáře.
- Překladač kodeků (Codec Translator) - používá moduly kodeků pro kódování a dekódování různých zvukových kompresních formátů používaných v telefonním prostředí. Množství dostupných kodeků je vhodné pro různorodé potřeby a docílení stavu rovnováhy mezi zvukovou kvalitou a použitou šířkou pásma.
- Scheduler a I/O manažer (Schedule and I/O manager) - ovládání rozvrhování nízkourovňových úloh a systémového řízení pro optimální výkon podle stavu zatížení.



Obr. 2: Blokové zobrazení architektury

Zaváděné moduly API:

Pro zaváděné moduly jsou definovány čtyři API, usnadňující oddělení hardwaru a protokolů. Použitím tohoto zaváděného modulového systému si jádro Asterisku nemusí dělat starosti s detaily o skutečnosti, jak se volající připojí, co používá za kodeky, atd.

- Kanálové API - ovládá typ spojení příchozího volání, tedy jedná-li se o VoIP spojení, ISDN, PRI, Robbed bit signaling nebo nějakou další technologii. Dynamické jednotky jsou zavedené pro ovládání detailů nižších vrstev těchto spojení.
- Aplikační API - aplikační API bere v úvahu různorodé jednotky úkolů, které mají běžet za účelem vykonávání různorodých funkcí. Conferencing, Paging, Výpis adresáře, Hlasová pošta, Přímý přenos dat a každý další úkol, který by PBX systém mohl vykonávat (teď nebo v budoucnu) je ovládán prostřednictvím těchto oddělených modulů.
- API překladače kodeků - zavádí moduly kodeků pro podporu různých audio formátů kódování a dekódování jako GSM, μ - law, A - law a také MP3.
- API souborových formátů - ovládá čtení a zápis různých souborových formátů pro ukládání dat v souborovém systému.

Používáním těchto API, Asterisk docílí kompletního oddělení mezi jádrem, pracujícím jako PBX server systém a různorodými existujícími technologiemi (nebo vyvíjenými). Modulární forma dovolí systému hladkou integraci a to jak aktuálně implementovaného hardwaru přepojované telefonie, tak rychle rostoucí hlasové technologie na bázi paketů. Schopnost zavádět moduly kodeků, umožní podporovat extrémně kompaktní kodeky, nutné pro paketový hlas přenášený přes pomalá spojení (např. telefonní modem) za předpokladu, že stále poskytují vysokou kvalitu zvuku přes méně omezená spojení.

Aplikační API poskytuje pružné použití aplikačních modulů pro vykonávání každé funkce a to flexibilně na požádání. Bere v úvahu otevřený rozvoj nových aplikací za účelem přizpůsobení unikátních uživatelských potřeb a situací. Navíc zavádění všech aplikací jako samostatných modulů umožní velmi pružný systém, dovolující administrátorovi navrhovat nejlepší vhodnou cestu pro volající na PBX systému a modifikovat směry volání vhodné pro změnu komunikačních potřeb výnosného podniku.

3. Dialplan – číslovací plán

Dialplan je konfigurován v souboru `extensions.conf`. Je to nejdůležitější konfigurační soubor systému. Řídí způsob ovládání a směrování příchozích a odchozích hovorů. Toto je místo, kde kontrolujete chování všech spojení provedených prostřednictvím PBX.

3.1 Organizace dialplan

3.1.1 [general]

V této sekci definujete některá obecná nastavení. Na začátku konfiguračního souboru by měla být sekce s označením:

```
[general]
```

, kde definujete několik hlavních voleb, týkajících se dialplan.

- `static`: v tomto stupni tato volba ovlivňuje pouze operace příkazů `save dialplan`. Default hodnota je `no`, ale vzorový soubor `extension.conf` instalovaný explicitně nastaví `static = yes`.
- `writeprotect`: Jestliže je nastaveno `writeprotect=yes` a `static=yes`, pak můžete ukládat aktuální `dialplan` prostřednictvím CLI (Command Line Interface) příkazu „`save dialplan`“. (definice globálních proměnných v sekci `[globals]` zůstane nezměněna). Default hodnota je `no`.

Upozornění:

- „`save dialplan`“ přepíše váš existující soubor `extensions.conf` nově generovaným z aktuálního `dialplan`. Kopie vašeho starého souboru `extensions.conf` nebude ponechána. Všechny vaše komentáře v souboru budou ztraceny.
- Neplatí skutečnost, že „`no`“ je default hodnota jestliže není nic specifikováno, ale explicitně instalovaný soubor `extensions.conf` nastavuje hodnoty `writeprotect=no` a `static=yes`. Tato skutečnost umožňuje zranitelnost souboru `extensions.conf` v default instalaci.

3.1.2 `[globals]`

Definice globálních proměnných `dialplan` a jejich počáteční hodnoty následují `[general]` nastavení. Sekce globálních proměnných začíná hlavičkou:

```
[globals]
```

V současnosti nejsou globální proměnné Asterisk obvykle používány jako proměnné, ale jako konstanty. Obvykle jsou použity pro možnost mít samostatné místo v `dialplan`, kde můžete specifikovat hodnoty, které máte možnost v budoucnu změnit, jestliže budete chtít změnit konfiguraci PBX.

Aktuální hodnota globální proměnné může být v `dialplan` změněna použitím příkazu `SetGlobalVar`, může být uvnitř `dialplan` vyjádřena syntaxí:

```
/${VariableName}
```

Poznámka: Jména globálních proměnných nejsou citlivé na velikost písmen, tedy `/${MYGLOBAL}` a `/${myGLObal}` jsou ekvivalentní.

Příklad: Definování některých globálních proměnných.

```
[globals]
```

```
; Určení které extension budou vyzvánět při příchozím volání
```

```
    INCOMING => Zap/3&Zap/4
```

```
; Určení jak dlouho budou tyto extension vyzvánět než volání místo toho přejde do voicemail
```

```
    RINGTIME => 3
```

```
; Určení jaký zvukový soubor bude přehrán jako úvodní pro voicemail
```

```
    VMANNOUNCE => mysounds/my-vm-announce
```

```
; Definování kanálů na které jsou jednotlivé extension připojeny
```

```
    KITCHEN => Zap/3
```

```
    STUDY => Zap/4
```

```
    HALL => Zap/5
```

```
; V případě že chceme provést odchozí volání, které linky použijeme
```

```
    OUTGOING => Zap/1&Zap/2
```

Vytvoření těchto definicí ve skutečnosti neprovádí vůbec nic. Systém ve skutečnosti například ví co má dělat s proměnnou nazvanou „INCOMING“. Je na vás definovat požadované proměnné a poté je použít v dialplanu požadovaným způsobem.

V tomto případě byly všechny proměnné napsány s použitím velkých písmen, ačkoliv globální proměnné nejsou citlivé na velikost písma. Použití velkých písmen je pouze konvencí pro odlišení od proměnných kanálů, které jsou obvykle psány ve smíšeném tvaru.

3.1.3 Používání proměnných

Asterisk může používat globální nebo kanálově specifické proměnné jako argumenty příkazů. Proměnné jsou odkazovány použitím následujícího syntaxu:

```
${foo}
```

Kde „foo“ je jméno dané proměnné. Jméno proměnné může být libovolný alfanumerický řetězec začínající písmenem. Uživatelsky definovaná jména proměnných nejsou citlivá na velikost písma. Oproti tomu proměnné definované Asteriskem (globální) jsou na tuto velikost citlivá. Například `${EXTEN}` bude funkční, ale `${exten}` nebude.

V systému Asterisk existují tři typy proměnných: globální proměnné, kanálové proměnné a proměnné prostředí.

- *Globální proměnné* mohou být nastaveny buď v části `[globals]` souboru `extensions.conf` nebo použitím příkazu `SetGlobalVar`. Jakmile jsou definované, mohou být odkazovány kdykoliv jakýmkoliv kanálem.
- *Kanálové proměnné* jsou nastavovány použitím příkazu `SetVar`. Každý kanál obdrží vlastní prostor proměnných, takže neexistuje možnost kolize mezi rozdílným voláním a proměnná je automaticky zničena po zavěšení kanálu.
- *Proměnné prostředí* poskytují možnost přístupu k proměnným prostředí unix v Asterisk.

Jestliže definujete proměnnou prostředí se shodným jménem jako globální proměnnou, pak odkazy na toto jméno vrátí hodnotu kanálové proměnné. Například předpokládejme, že definujete context „FooTest“ s jednoduchým extension 100 s následující definicí:

```
[FooTest]
exten => 100,1,SetGlobalVar(FOO=5)
exten => 100,2,NoOp(${FOO})
exten => 100,3,NoOp(${foo})
exten => 100,4,SetVar(foo=8)
exten => 100,5,NoOp(${FOO})
exten => 100,6,NoOp(${foo})
```

Příkaz `NoOp` použijeme z důvodu pomoci zkoušení programu. Jestliže vytočíte extension 100 v context `FooTest` a máte spuštěný Asterisk s konzolou generující hlášení, můžete obdržet podobné hlášení systému:

- Executing `SetGlobalVar("Zap/1-1", "FOO=5")` in new stack
- Setting global variable 'FOO' to '5'
- Executing `NoOp("Zap/1-1", "5")` in new stack
- Executing `NoOp("Zap/1-1", "5")` in new stack

- Executing SetVar("Zap/1-1", "foo=8") in new stack
- Executing NoOp("Zap/1-1", "8") in new stack
- Executing NoOp("Zap/1-1", "8") in new stack

Zjistíte, že volání *SetGlobalVar*, `_${FOO}` a `_${foo}` vrátí hodnotu globální proměnné (konkrétně 5). Po volání *SetVar*, globální proměnná „foo“ byla zatemněna kanálovou proměnnou „foo“, poté `_${FOO}` a `_${foo}` dají hodnotu 8. Nicméně hodnota globální proměnné zůstává nezměněna (5) a každý jiný kanál odkazující na globální proměnnou `_${foo}` bude stále udávat hodnotu 5.

3.1.4 Proměnné a výrazy

Také existuje podpora pro používání proměnných ve tvaru `_${VARIABLENAME}`. Rovněž můžete použít výraz ve tvaru `_${EXPRESSION}`, kde může být výraz regulárním výrazem, porovnáním, přírůstkem, odčítáním atd. Pro důkladnější vysvětlení výrazů a proměnných se podívejte do Asterisk dokumentace.

3.1.5 Funkce manipulace s řetězci

Délka řetězce

`_${LEN(foo)}` - vrací délku řetězce nazvaného „foo“.

Příklad:

```
exten => 100,1,SetVar(Fruit=pear)
exten => 100,2,NoOp(${LEN(Fruit)})
exten => 100,3,NoOp(${LEN(${Fruit}}))
```

První NoOp by ukázal hodnotu 5 (délka řetězce „fruit“). Druhý NoOp by ukázal hodnotu 4 (délka řetězce „pear“). Toto je možnost jak kontrolovat NULL nebo prázdný řetězec.

Podřetězce

`_${foo:offset:length}`

- vrací podřetězec řetězce „foo“, začíná s daným offsetem a vrací další znaky

- Jestliže je offset záporný, vezme pozici doleva od pravého konce řetězce
- Jestliže je délka vynechána nebo záporná, poté všechny zbývající části řetězce začínají s offsetem.

Příklad:

```
_${123456789:1} - vrací řetězec 23456789
_${123456789:-4} - vrací řetězec 6789
_${123456789:0:3} - vrací řetězec 123
_${123456789:2:3} - vrací řetězec 345
_${123456789:-4:3} - vrací řetězec 678
```

Příklad použití:

```
exten => _NXX.,1,SetVar(areacode=${EXTEN:0:3})
- obdrží první 3 čísla z ${EXTEN}
```

```
exten => _516XXXXXXXX,1,Dial(${EXTEN:3})  
    - obdrží všechny ale první 3 čísla z ${EXTEN}  
exten => 100,1,SetVar(whichVowel=4)  
exten => 100,2,SetVar(foo=AEIOU:${whichVowel}:1)  
    - nastaví ${foo} na samostatné písmeno „U“
```

Zřetězení řetězců

Jednoduché zřetězení dvou řetězců:

```
${foo}${bar}  
555${theNumber}  
${longDistancePrefix}555${theNumber}
```

3.2 Context a extension

Číslovací plán se skládá ze souboru několika context. Tyto definované context jsou nejdůležitější částí konfiguračního souboru extensions.conf a jsou jednou z nejdůležitějších částí celé konfigurace systému. Context je pouze souborem několika extensions.

Příklad:

```
Context "mainmenu":  
Extension Description  
s          Uvítací zpráva a instrukce  
1          Prodej  
2          Podpora  
3          Účtování  
9          Adresář  
#          Zavěšení
```

Tento context s názvem „mainmenu“ má pouze jedno číselnou volbu. Bod extension „s“ je počátečním extension kde volající začíná. Tento extension může například přehrát uvítací zprávu. Pro volbu Prodej stiskněte 1, pro Podporu stiskněte 2, pro Účtování volte 3, volba 9 je pro adresář společnosti, případně volte „#“ pro zavěšení. Ve skutečnosti může každá tato položka buď vytvořit něčí skutečné extension nebo může volajícího směřovat do dalšího menu.

Context mohou být používány za účelem implementování čísel důležitých rysů, jako například:

- Security - povolení mezinárodních volání pouze z určitých telefonů
- Routing - směrování hovorů na základě extension
- Autoattendant - uvítání volajících a vyzvání k volbě extension
- Multilevel menus - menu pro prodej, podporu atd.
- Authentication - dotaz na přístupové heslo pro určité extension
- Callback - redukce mezinárodních poplatků
- Privacy - seznam nežádoucích volajících čísel na dané extension
- PBX Multihosting - lze vytvářet „virtuální hosty“ na PBX
- Daytime/Nighttime - lze měnit chování na základě hodin
- Macros - vytváření skriptů pro běžně používané funkce

Když Asterisk obdrží na kanále příchozí spojení, podívá se do context definovaného pro konkrétní kanál a tím určí co má systém provést. Context definuje různé soubory příkazů závislejících na skutečnosti, kterou extension uživatel vytočil. Například může context poskytnout jeden soubor příkazů pro stav, kdy uživatel vytočil „123“ a jiný soubor příkazů pro stav, kdy uživatel vytočil „9“ nebo třeba číslo začínající „555“.

Pro některé druhy spojení (například příchozí volání z venkovní telefonní linky) nevolí uživatel žádnou extension. V tomto případě se systém zachová jakoby uživatel volil speciální extension nazvaný „s“ (jako Start). Asterisk hledá extension se jménem „s“ v definicích context kanálu, kterému jsou instrukce určeny pro ovládání volání.

Pro příklad předpokládejme, že máte kanál „Zap/1“ spojený s telefonním přístrojem v budově. Dále předpokládejme, že v konfiguračním souboru pro Zap kanály (zapata.conf) máte definován context = john pro Zap kanál 1. Takže, použijeme-li telefonní přístroj pro vytočení čísla, Asterisk se podívá na context se jménem „john“ za účelem zjištění co má v tomto případě provést. Definici v souboru extensions.conf začínáte prostřednictvím vložení jména context do hranatých závorek na samostatném řádku, například:

[john]

Pro každý context potřebujete definovat jednu nebo více extension, které Asterisk používá pro porovnávání s vytáčeným číslem. Pro každý extension musíte také systému definovat skutečnost, co má v daném případě provádět a to prostřednictvím souboru příkazů.

3.2.1 Používání context

Když obdrží Asterisk volání příslušející k dané extension (buď příchozí volání z venkovní nebo z vnitřní extension). Určení kterému context volání přísluší, závisí na skutečnosti, kterým kanálem volání přišlo. Konfigurujete-li kanály prostřednictvím Asterisk PBX, jednou z věcí kterou musíte provést je definovat do kterého context bude příchozí volání na kanále umístěno. K tomuto lze použít následující definici:

```
context=incoming
```

Prvním způsobem jak používat context je přinutit Asterisk provádět rozdílné věci v závislosti na skutečnosti odkud volání přichází. Určitě budete mít definován více než jeden context pro to, jak systém ovládá příchozí volání na Vaši telefonní linku:

- může vyzvánět některý z Vašich extension
- může přehrávat zprávu
- může nahrávat voicemail zprávu
- a tak dále ...

Asterisk můžete používat pro ovládání spojení z vnitřních extension - může být povoleno vytáčení různých extension nebo provádění odchozího hovoru - tímto způsobem definujete skutečnost, že volání z různých kanálů směřuje do různých context.

3.2.2 Začlenění context

Jeden extension context může obsahovat jiný. Například uvažujme následující contexty:

Context "default":

Extension	Description
101	Mark Spencer
102	Wil Meadows
0	Operator

Context "local":

Extension	Description
_9NXXXXXX	Local calls

include => "default"

Context "longdistance":

Extension	Description
_91NXXNXXXXXX	Long distance calls

include => "local"

Máme definovány následující contexty:

- Context *default* - povoluje volání tří telefonních extension (Mark, Wil a Operator).
- Context *local* - obsahuje jeden vzor extension pro povolení vytáčení pouze 7 číselné volby (místní volání) a také obsahuje context „default“, tedy uživateli zpřístupní volání na extension Mark, Wil a Operator.
- Context *long distance* - má jeden vzor extension pro přístup k meziměstským volání a obsahuje context „local“ což umožňuje místní volání a také volání na extension Mark, Wil a Operator.

Použitím vhodných kontextů extension můžete bezpečně kontrolovat skutečnost kdo má přístup k placeným službám.

3.3 Extensions

Extension může být ve dvou typech: *prostý* nebo *vzor*.

Prostý extension může být číslo (např. 1234) a může také obsahovat standardní symboly * a #, které se objevují na běžných telefonech (např. 12#34*56 je platná extension). Některé telefonní klávesnice mají také speciální DTMF tlačítka označované A, B, C a D, extension může být poté definována i prostřednictvím těchto písmen. Ve skutečnosti může jméno extension obsahovat každé písmeno nebo číslici, právě tak jako některé interpunkční značky. Některé VoIP telefony jsou schopny „vytáčet“ extension „čísla“, které mohou mít tvar libovolného řetězce (např. „Office“). V Asterisku je také možné definovat extension ve tvaru „Office“.

Odpověď na otázku zda jsou jména extension citlivá na velikost písma není zcela jednoznačná. Na velikost písma jsou citlivá v případě, že Asterisk zkouší porovnat extension vytočené uživatelem proti extension definovanému pro context. V tomto případě musí obě extension souhlasit a to včetně velikosti písma. Takže jestliže uživatel prostřednictvím VoIP telefonu vytočí „OFFICE“, systém nebude provádět příkazy definované pro extension „Office“. Na druhou stranu nejsou jména extension citlivá na velikost písma v případě, že nedefinujete rozdílnou extension (ve stejném context), která má stejné jméno lišící se pouze ve velikosti písma. Tedy nemůžete definovat jeden soubor příkazů pro extension „Office“ a jiný soubor příkazů pro extension „OFFICE“.

3.3.1 Jména Extensions

Extension Dialplan mohou být jednoduchá čísla jako „421“ nebo „0“. Mohou to být také alfanumerická jména jako „james“ nebo „bond007“. Třebaže typický telefon není schopen volit extension nazvaný „james“ (některé i toto umožňují), často může Váš logický Dialplan zahrnovat skoky z jedné extension do jiné extension a pro tyto skoky můžete definovat extension jména s jakýmkoliv jménem chcete, ačkoliv si nepřejete je vytáčet přímo.

Samozřejmě tlačítkové telefony nemají pouze číslice 0 – 9, mají také znaky „*“ a „#“. Některé tlačítkové telefony mají také čtyři extra číselné znaky (A, B, C a D). Jestliže máte v organizaci takovýto handset, nic vám nebrání vytvořit možnost použití těchto dodatečných tlačítek pro vlastní speciální účely.

Poznámka: Pro vytvoření extension, které je spouštěno volení symbolu „#“ musíte použít vzor extension. Asterisk nerozeznává symbol „#“ jako běžný číselný znak, třebaže se tento objevuje na všech telefonech s volbou DTMF.

3.3.2 Předdefinovaná jména extension

Pro některé speciální účely používá Asterisk následující extension:

i :	Invalid	neplatný
s :	Start	začátek
h :	Hangup	zavěšení
t :	Timeout	překročení časového limitu
o :	Operator	operátor

3.3.3 Definování extension

Na rozdíl od tradiční PBX, kde jsou extension asociovány s telefonem, rozhráním, menu atd., jsou v Asterisk extension definovány jako seznam příkazů určených k provedení. Příkazy jsou obecně vykonávány v pořadí v jakém jsou definovány prostřednictvím svých prioritních tag. Přesto však některé příkazy (jako příkazy *Dial* a *GotoIf*) mají schopnost přesměrování na jiné místo na základě definované podmínky.

V případě, že je vytočena extension, je proveden příkaz označený prioritou 1, následně příkaz s prioritou 2 atd. Toto je prováděno dokud nenastane některý z následujících stavů:

- Volání je zavěšeno
- Příkaz vrátí kód výsledku –1 (indikace poruchy)
- Příkaz s některou vyšší prioritou neexistuje (poznamenejme, že systém nepřeskočí chybějící prioritu)
- Volání je směrováno na novou extension

V syntaxi souboru `extensions.conf` je každý krok příkazu v extension napsán v následujícím formátu:

```
exten => pattern,priority,Command(parameters)
```

Syntax pro definování extension v Asterisk konfiguračním souboru `extensions.conf` je spíše těžkopádná. Naštěstí je zde několik pomocných nástrojů (GUI tool).

Definice jednotlivé extension obsahuje jeden nebo více příkazů. Každý tento příkaz je uveden na samostatném řádku v následujícím formátu:

```
exten => extension,priority,Command(parameters)
```

- *extension* je jméno extension, ať už písmenné jméno nebo vzor extension. Toto jméno je přesně opakováno pro každý další příkaz s nižší prioritou pro dané extension.
- *priority* je číslo typu Integer. Pojmenování „priorita“ je částečně zavádějící, protože se ve skutečnosti jedná pouze o číslo příkazu příslušejícímu k extension. Vlastní provádění příkazů začíná systém na místě priority 1. Jestliže se řádek s prioritou 1 nevyskytuje, pak extension nebude odpovídat žádnému volenému číslu. Jestliže po řádku s prioritou 1 není řádek s prioritou 2, pak Asterisk ukončí provádění pro danou extension a to i v případě, že jsou dále uvedeny příkazy s prioritou 3 nebo více. Nicméně při provádění některých příkazů může systém provést skok (v závislosti na logice příkazu) na jinou prioritu, než která logicky následuje.
- *command* je jméno prováděného příkazu (také nazýváno „application“). Více v seznamu Asterisk příkazů.
- *parameters* závisí na daném příkazu. Některé příkazy tyto parametry nevyžadují, lze je vynechat.

Příklad:

```
exten => 123,1,Answer
exten => 123,2,Playback(tt-weasels)
exten => 123,3,Voicemail(44)
exten => 123,4,Hangup
```

Toto je definice jednoduché extension nazvané „123“. V případě, že je prováděno volání na extension „123“, Asterisk na volání odpoví sám, přehraje zvukový soubor nazvaný „tt-weasels“, poskytne uživateli možnost zanechat voicemail zprávu v mailbox (44) a poté zavěsí.

Asterisk se nestará o pořadí ve kterém vkládáte řádky do souboru extensions.conf. Můžete míchat řádky v různém pořadí. Tato skutečnost však na funkci nemá vliv, protože systém pro určení skutečnosti, který příkaz má provést, použije prioritu.

Další volby pro definování extension obsahují volby obvykle označovány jako „logika ex-příteřkyně“. Tato logika odpovídá volené extension, jestli přichází zvenku nebo zevnitř, na základě callerID volající osoby.

Příklad:

```
exten => 123/100,1,Answer()
exten => 123/100,2,Playback(tt-weasels)
exten => 123/100,3,Voicemail(123)
exten => 123/100,4,Hangup()
```

Toto odpovídá extension 123 a vykoná následující volby POUZE je-li Caller-ID Number volajícího účastníka 100. Toto může být také uděláno odpovídajícím vzorem, jak je vidět dále:

```
exten => 1234/_256NXXXXXX,1,Answer()
```

Toto odpovídá pouze pro extension 1234 a to jestliže Caller ID Number začíná číslem 256. Tento tvar je velmi užitečný pro zachování locals pro vytáčení bezplatných čísel a nákladů za volání.

Příklad:

```
exten => s,1,Answer
```

```
exten => s/9184238080,2,SetCIDName(EVIL BASTARD)
exten => s,2,SetCIDName(Good Person)
exten => s,3,Dial(SIP/goodperson)
```

V podstatě volání přichází, v prioritě 2 oddělíte osoby které nemáte rádi. Kdokoliv jiný zůstává ve správném směru a prioritě 3 vrací každého do hlavního směru.

3.3.4 Vzory Extension

Jména extension nejsou tedy omezena na jednotlivé specifické extension „čísla“. Jednotlivé extension mohou rovněž odpovídat vzorům. V souboru extension.conf se jméno extension stává vzorem, jestliže začíná znakem „podtržítka“. V extension vzorech mají následující znaky speciální smysl:

X odpovídá některé z číslic 0 – 9
Z odpovídá některé z číslic 1 – 9
N odpovídá některé z číslic 2 – 9
[1237-9] odpovídá některé číslici nebo písmenu uvedenému v závorkách (v tomto případě tedy 1,2,3,7,8,9)

Příklad:

Context "routing":

Extension	Description
_61XX	Dallas Office
_62XX	Dallas Office
_7[1-3]XX	San Jose Office
_7[04-9]XX	Los Angeles Office

Tento context nazvaný „routing“ posílá volání na různé servery podle jejich extension. Tato společnost rozhodla, že všechny telefonní extension budou dlouhé 4 čísla. Jestliže uživatel volá extension začínající 61 nebo 62 bude směřován na Dallas. Vše začínající na 71, 72 nebo 73 bude směřováno do San Jose a všechno zbývající (70, 74-79) bude směřováno na Los Angeles.

Příklad:

_NXXXXXX	odpovídá obvyklému 7 číselnému telefonnímu číslu
_1NXXNXXXXXX	odpovídá kódu oblasti a telefonnímu číslu předcházejícímu jedničkou
_9011.	odpovídá libovolnému řetězci pěti znaků začínajících 9011, neznamená to však vlastní znakový řetězec 9011
_#	odpovídá jednomu stisknutí tlačítka „#“

Upozornění: Nepoužívejte vzor „_“. Protože tento tvar znamená všechno a to včetně speciálních Asterisk extension jako „i“, „t“, „h“ atd. Místo toho použijte například „_X.“ nebo „_X“ což nemá význam speciálních extension.

3.4 Pomocné funkce

3.4.1 Znovu zavedení (Reloading)

Pokud požadujete opětovné zavedení dialplan po provedených změnách a to bez reloadu celého Asterisk, použijte Asterisk CLI příkaz *extensions reload*.

3.4.2 Problematika rozdělení souboru

Pomocí příkazu *#include <filename>* v souboru *extensions.conf* jsou do tohoto souboru zahrnuty další soubory. Tímto způsobem můžete nastavit systém, kde je *extensions.conf* pouze hlavním souborem, ale ne jediným. V dalším kroku můžete vytvořit například soubor *users.conf* obsahující lokální uživatele, soubor *services.conf* obsahující různé služby, atd. Touto cestou může být dialplan lépe a snadněji udržován a to bez závislosti na velikosti vašeho nastavení. Příkaz *#include <filename>* není shodný s příkazem *include <context>*. Příkaz *#include* pracuje ve všech Asterisk konfiguračních souborech.

3.4.3 Forwarding na jiný Asterisk

Syntax:

```
[iaxprovider]
switch => IAX2/user:[key]@server/context
```

Specifikuje forwarding na jiný Asterisk. Potřebný *user* a *key* budou definovány v souboru *iax.conf* volaného serveru. Použije se *context* v souboru *extensions.conf* volaného serveru.

3.4.4 Ovládání *extensions.conf* zvenku

- Asterisk extension z *mysql*
- Jako u všech souborů typu **.conf* můžete použít příkaz *#include* pro zahrnutí jiného souboru

Příklad: Jednoduchý *extensions.conf* použitím příkazu *#include*:

```
#include "my-extra-config-file"
```

```
[globals]
ALL=Zap/1&SIP/1000&SIP/1001
```

```
[default]
exten => s,1,Answer
exten => s,2,Playback(welcome-message)
exten => s,3,Goto(context-in-include-file,s,1) ; go to context defined in included file
...
```

3.4.5 Použití makra pro vytvoření extension

Příklad:

```
[globals]
PHONE1=Zap/1
PHONE2=SIP/6002

[macro-oneline]
exten => s,1,Dial(${ARG1},20,t)
exten => s,2,VoiceMail(u${MACRO_EXTEN})
exten => s,3,Hangup
exten => s,102,VoiceMail(b${MACRO_EXTEN})
exten => s,103,Hangup

[local]
exten => 6601,1,Macro(oneline,${PHONE1})
exten => 6602,1,Macro(oneline,${PHONE2})
```

3.5 Předdefinované kanálové proměnné

Existují některé kanálové proměnné nastavené prostřednictvím systému, které můžete použít v definici vašeho dialplan. Tyto proměnné jsou, oproti uživatelem definovaným proměnným, citlivé na velikost použitého písma.

- `${ACCOUNTCODE}`: Kód účtu, jestliže je definován (viz. Asterisk billing)
- `${ANSWEREDTIME}`: Čas kdy bude volání odpovězeno.
- `${CALLERID}`: Aktuální Caller ID (jméno a číslo)
- `${CALLERIDNAME}`: Aktuální Caller ID jméno
- `${CALLERIDNUM}`: Aktuální Caller ID číslo
- `${CALLINGPRES}`: Proměnná PRI Call ID Presentation pro příchozí volání
- `${CHANNEL}`: Aktuální jméno kanálu
- `${CONTEXT}`: Jméno aktuálního context
- `${DATETIME}`: Aktuální datum a čas ve formátu DDMMYYYY-HH:MM:SS
- `${DIALEDPEERNAME}`: Jméno volané strany. V současnosti „rozbité“, viz. DIALEDPEERNAME
- `${DIALEDPEERNUMBER}`: Číslo volané strany. V současnosti „rozbité“, viz. DIALEDPEERNUMBER
- `${DIALEDTIME}`: Čas kdy bylo číslo voleno.
- `${DIALSTATUS}`: Stav volání. Viz. DIALSTATUS
- `${DNID}`: Identifikátor voleného čísla (Dialed Number Identifier). Limitované použití viz. DNID
- `${EPOCH}`: Aktuální UNIX-style epocha (číslo v sekundách od 1 Jan 1970)
- `${EXTEN}`: Aktuální extension
- `${HANGUPCAUSE}`: Poslední návratová kód zavěšení na Zap kanále připojeném k PRI rozhraní
- `${INVALID_EXTEN}`: Žádané extension při přesměrování na „i“ (invalid) extension
- `${LANGUAGE}`: Aktuální jazykové nastavení

- `MEETMESECS`: Číslo v sekundách určující jak dlouho se uživatel účastnil MeetMe konference
- `PRIORITY`: Aktuální priorita
- `RDNIS`: Aktuální přesměrování DNIS, Caller ID které přesměrovalo volání. Limitované použití viz. RDNIS
- `SIPDOMAIN`: SIP destination domain volání v pásmu (inbound)
- `SIP_CODEC`: Použito pro nastavení SIP kodeku pro volání (patrně „rozbité“ ve verzi 1.0.1, v pořádku verze 1.0.3 & 1.0.4, nevíme o verzi 1.0.2)
- `SIPCALLID`: SIP dialog Call-ID: záhlaví
- `SIPUSERAGENT`: SIP user agent záhlaví
- `TIMESTAMP`: Aktuální datum a čas ve formátu YYYYMMDD-HHMMSS
- `TXTCIDNAME`: Výsledek aplikace TXTCIDName
- `UNIQUEID`: Aktuální unikátní identifikátor volání (call unique identifier)
- `TOUCH_MONITOR`: použito pro „jedno dotekový záznam“ (viz. features.conf a wW dial flags). Jestliže je toto nastaveno na obou stranch volání, poté var obsahuje app_args pro app_monitor neboli je použit default WAV|m

3.6 Aplikačně specifické proměnné

Některé aplikace zabírají extra vstupy nebo poskytují extra výstupy používáním kanálových proměnných.

- ChanIsAvail vrací `AVAILCHAN`: První dosažitelný kanál
- Dial bere vstup z `VXML_URL`: Posílá XML Url telefonu Cisco 7960
- Dial bere vstup z `ALERT_INFO`: Nastavuje tempo vyzvánění Cisco telefonů
- Dial vrací `CAUSECODE`: Jestliže volba selže, toto je errormessage
- Dial vrací `DIALSTATUS`: Stav návratového textového kódu posledního pokusu vytáčení
- EnumLookup vrací `ENUM`: Výsledek vyhledávání
- MeetMe bere vstup z `MEETME_AGI_BACKGROUND`: AGI skript pro spuštění
- MeetMe vrací `MEETMESECS`: Počet sekund jak dlouho byl uživatel v konferenci
- Hangup čte `PRI_CAUSE` proměnná pronastavení návratových PRI kódů (PRI return codes)
- TXTLookup vrací `TXTCIDNAME`: Výsledek DNS vyhledávání
- AgentCallbackLogin vrací `AGENTBYCALLERID_${CALLERID}`: ID úspěšně přihlášeného agenta

3.7 Proměnné specifické pro makra

V případě, že použijete context makra, jsou dostupné extra kanálové proměnné.

- `ARG1`: První argument makra
- `ARG2`: Druhý argument makra (a tak dále)
- `MACRO_CONTEXT`: Context extension, která spouští toto makro
- `MACRO_EXTEN`: Extension, která spouští toto makro

- `ENV(MACRO_OFFSET)`: Nastaveno prostřednictvím makra za účelem ovlivnění priority, kde bude vykonávání pokračovat v případě opuštění makra
- `ENV(MACRO_PRIORITY)`: Priorita v extension, kde bude makro spuštěno

3.8 Proměnné prostředí unix

Prostřednictvím následující syntaxe můžete zpřístupnit proměnné prostředí unix:

```
ENV(foo)
```

- `ENV(ASTERISK_PROMPT)`: Aktuální Asterisk CLI prompt.
- `ENV(RECORDED_FILE)`: jméno posledního souboru uloženého prostřednictvím příkazu Record (dostupné v CVS > 2004-03-21)

4. Kanály Asterisku

4.1 Pojem kanál

Kanály jsou logická spojení s různými signalizačními a přenosovými cestami, které může Asterisk využívat k vytváření a spojování jednotlivých hovorů. Kanál by mohl představovat spojení s obyčejným telefonním přístrojem (telefonní linkou) nebo Internetové telefonní hovory („logické volání“). Asterisk nedělá žádný rozdíl mezi typem kanálu „FXO (Foreign eXchange Office)“ a „FXS (Foreign eXchange Station)“, nerozlišuje tedy mezi telefonními linkami a telefony. Každé volání je umístěno na odlišném kanále.

Prostřednictvím kanálů vstupují do systému různé formáty komunikace; fyzické telekomunikační okruhy (jako FXO, FXS, PRI, BRI), softwarově založené spojení, síťově připojitelné entity (jako SIP a IAX) a exkluzivní vnitřní kanály Asterisku určené pro všechny dodatečné prostředky (jako Agent, Console, Local). Asterisk se všemi těmito kanály zachází jako s přípojnými body, jejichž vzájemnou interakci provádíte v Dialplan (`extensions.conf`). Je důležité si uvědomit, že i kdyby se kanály lišily v rámci použité technologie a konektivity, Asterisk vám dovolí zacházet se všemi jakoby byly téměř stejné.

Díky způsobu zacházení s kanály je Asterisk extrémně flexibilní a silnou PBX. V mnoha klasických PBX mají různé kanály zcela jiné způsoby komunikace. Asterisk vám umožní jednat s daným kanálovým typem stejným způsobem jako s ostatními kanálovými typy.

4.2 Typy kanálů

Asterisk ve standardní distribuci poskytuje následující typy kanálů:

- Agent – ACD Agent kanál
- Console – konzolový klientský ovladač pro zvukové karty
- H.323
- IAX nebo IAX2 – Inter-Asterisk Exchange Protocol

- Local – Loopback do dalšího kontextu
- MGCP – Media Gateway Control Protocol
- Modem – pro připojení ISDN linek
- phone – Linuxový telefonní kanál
- SIP – Session Initiation Protocol
- Skinny – ovladač pro Cisco Skinny Client Control Protocol
- VOFR – Voice over Frame Relay
- VPB – pro připojení obyčejného telefonu a telefonní linky používající Voicetronix karty
- Zap – pro připojení obyčejného telefonu a telefonní linky používající Digium karty

Ovladače kanálu nabízejí další technologie, které mohou být volitelně instalovány:

- bluetooth – dovolí používat bluetooth zařízení pro změnu směrování
- CAPI – ISDN CAPI kanál
- mISDN – mISDN kanál
- SCCP – alternativní Skinny/SCCP kanál
- UNISTIM – Nortel UNISTIM kanál

4.1 Kanál Agent

chan_agent je pseudo-kanál pro směrování volání k ACD agentovi. Agenti jsou definováni v agents.conf a aktivováni speciální log-in procedurou. To dovolí agentovi přizpůsobit se, aby byl k dispozici pro odebrání hovoru z fronty pomocí přihlášení se k telefonu. Agent může později zastavit odebrání hovorů pomocí odhlášení se od telefonu. Agenti mají smysl pouze v ACD uspořádání. Normálně jsou definováni jako účastníci fronty (hovory čekající ve frontě) uvnitř queues.conf.

4.2 Kanál H.323

Asterisk H.323 kanál je obsažen ve zdrojové distribuci Asterisku v adresáři channels/h323 zdrojového stromu. Kanál chan_h323 jedná pouze jako H.323 gateway, ne však jako gatekeeper.

Další implementací je H.323 kanálu nazvaný *asterisk-oh323*, která je aktivně vyvíjena v projektu InAccess Network (<http://www.inaccessnetworks.com/projects/asterisk-oh323>).

Brzy bude pravděpodobně do Asterisk systému implementován kanál nazvaný *asterisk-ooh323c*. Tento je založen na open sorce H.323 firmy Objective systems (<http://obj-sys.com/open>). Projekt je založen na implementaci H.323 signalizačního kanálu a předpokladu, že zpracování bude provádět samotný Asterisk.

4.3 Kanál IAX

IAXtel, služba nabízená firmou Digium, od prosince 2003 podporuje pouze protokol IAX2. Protokol IAX je od ledna 2004 totožný s protokolem IAX2.

IAX2 může být použit pro:

- vzájemné spojení Asteriskových serverů
- připojení k službě IAXtel
- spojení IAX klientů

Konfigurace kanálů se provádí modifikací souboru iax.conf.

4.4 Kanál Local

Kanál chan_local je pseudo-kanál. Používá se pro vytvoření smyčky, která volá zpět do Dialplan v různých context. Užitečné je především rekurzivní směrování, které je schopno vracet se do Dialplan po ukončení volání.

4.5 Kanály Modem

Kanál Modem je použit pouze pro ISDN karty ovládané i4l (isdn4linux) ovladačem. Existuje zde také alternativa této metody, která používá ISDN hardware uvnitř Asterisku (použití CAPI kanálu - chan_capi).

4.6 Kanálový modul SIP

SIP kanálový modul umožní Asterisku VoIP komunikaci se SIP telefony a ústřednami. Asterisk může působit jako:

- SIP klient – to znamená, že se Asterisk zaregistruje jako klient k jinému SIP serveru a přijímá a umísťuje volání k tomuto serveru.
- SIP server – Asterisk může být nakonfigurován tak, že se SIP klienti (telefony, softwaroví klienti) zaregistrují k serveru Asterisk a sestaví SIP relaci se serverem.
- SIP gateway – Asterisk vystupuje jako media gateway mezi SIP, IAX, MGCP, H.323 a PSTN spojením.

Konfigurace SIP kanálů se provádí modifikací souboru sip.conf.

4.7 Kanál vpb

Jestliže máte Voicetronix kartu rozhraní (např. OpenLine4, OpenSwitch8, OpenSwitch16) budete používat jejich linuxové ovladače. Asterisk komunikuje s těmito ovladači použitím vpb kanálového modulu. Konfigurace vpb kanálového modulu se provádí modifikací souboru vpb.conf.

4.8 Zap kanálový modul

Zap kanálový modul poskytne mezivrstvu (interface layer) mezi Asteriskem na jedné straně a Zaptel a/nebo ZapHFC ovladači rozhraní na straně druhé. Tyto ovladače poskytují schopnost používat karty rozhraní pro připojení vaší PBX k tradičnímu digitálnímu nebo analogovému telefonnímu vybavení:

```
Asterisk <--> ZapModule <--> ZaptelDriver <--> Digium Interface card <--> Phone/switch/PSTN
ZapHFCDriver <--> ISDN Card with HFC-Chipset <--> PSTN
```

Konfigurace ZAP kanálů se provádí modifikací souboru zapata.conf.

5. Instalace Asterisku

Instalace Asterisku byla provedena prostřednictvím následujícího příkazu. Tímto byla nainstalována základní distribuce Asterisku společně s konfiguračními soubory a soubory určené pro vývoj.

```
apt-get install asterisk asterisk-config asterisk-dev
```

Dodatečně byla doinstalována podpora pro vzájemné propojení s jinou SW ústřednou Asterisk pomocí protokolu IAX (Inter-Asterisk Exchange).

```
apt-get install libiax-dev libiax0
```

Tímto je Asterisk nainstalován a po nakonfigurování prostřednictvím úpravy konfiguračních souborů je možno jej spustit prostřednictvím následujícího příkazu, který spustí Asterisk v tzv. konzolovém módu.

```
asterisk -vvvgc
```

Poznámka:

Z důvodu rozšíření dostupných zvukových souborů pro vytváření hlasových menu, přehrávání audio odpovědí apod., byly doinstalovány soubory prostřednictvím CVS.

```
cd /usr/src
export CVSROOT=:pserver:anoncvs@cvs.digium.com:/usr/cvsroot
cvs login ; the password is anoncvs.
cvs checkout asterisk-sounds
cd asterisk-sounds
make install
```

6. Konfigurační soubory

Uvedené konfigurační soubory vychází s testovací instalace Asterisku na VŠB-TU Ostrava. V následujícím textu jsou rozebrány pouze některé důležité parametry, ostatní jsou uvedeny buď v teoretické části předchozího textu nebo v dokumentaci.

6.1 Dialplan (extensions.conf)

```
; Testovací Dialplan (extensions.conf)

[general]

static=yes
writeprotect=no ; zamezení prepisovani diaplan z CLI
```

```
[globals]

OPERATOR = SIP/421           ; definovany kanal/telefonni cislo
SUPPORT = SIP/424           ; odkazovani na promennou ${OPERATOR}

; Cesty pro dodatecne sound files definovane jako promenne
sounds_exten = /var/lib/asterisk/sounds/
sounds_phonetic = /var/lib/asterisk/sounds/phonetic/
sounds_letters = /var/lib/asterisk/sounds/letters/
sounds_silence = /var/lib/asterisk/sounds/silence/

; Urceni prefixu pro dany protokol - volba
; SIP_PREFIX = _42
; H323_PREFIX = _43
; SKINNY_PREFIX = _44
; IAX2_PREFIX = _45

TUO_TRUNK = IAX2/tuo:testwest@tuo
```

Proměnná TUO_TRUNK definuje propojení s další ústřednou Asterisk. Je použit protokol IAX2, přístupové jméno, heslo a server. Parametry uvedeného serveru jsou definovány v souboru iax.conf (viz. dále).

```
[macro-basic]

exten => s,1,SetCallerID(${CALLERID})
exten => s,2,SetCIDName(${CALLERNAME})
exten => s,3,Dial(${ARG1})
exten => s,4,Congestion
exten => s,5,Hangup
```

Makro nazvané „Basic“ definuje základní prováděné akce, které jsou provedeny při vytvoření daného čísla. Je dále použito při definování konkrétních extension. Pomocí příkazů nastavujeme základní popis volání (CallerID, CallerName). Následně je volba vytvořena, v případě obsazení je signalizováno obsazení.

```
[skupina]

; extension SIP
exten => _421,1,Macro(basic,SIP/421) ; Zuka
exten => _424,1,Macro(basic,SIP/424) ; Wija
exten => _428,1,Macro(basic,SIP/428) ; test SW s cryptovanim md5
exten => _429,1,Macro(basic,SIP/429) ; test SW

exten => _450,1,Macro(basic,IAX2/450) ; Ethernet Phone AT-320

exten => _497,1,Macro(basic,SIP/497) ; SW phone Messenger
exten => _498,1,Macro(basic,SIP/498) ; HW phone Grandstream Budge Tone 100
```

Kontext „skupina“ používáme k definování telefonních čísel. Použijeme předem definované makro „basic“ s parametrem kanál/číslo (SIP/xxx) respektive (IAX2/xxx). Znak „podtržítka“ označuje skutečnost, že čísla za tímto znakem jsou volena přímo.

```
; smerovani na cislo, které není definovane v sip.conf  
exten => _420,1,Macro(basic,${TUO_TRUNK}/${EXTEN})
```

Pro směrování volání na telefonní číslo, které je definováno v jiném Asterisku použijeme předcházející tvar. Předpokládáme, že uvedené číslo „420“ je definováno v příslušných konfiguračních souborech jiného Asterisku (extensions.conf, sip.conf).

```
; „echotest“ – „Bravo Charlie Tango is not available“  
exten => _499,1,Answer()  
exten => _499,2,Playback(${sounds_phonetic}bravo)  
exten => _499,3,Wait(1)  
exten => _499,4,Playback(${sounds_phonetic}charlie)  
exten => _499,5,Wait(1)  
exten => _499,6,Playback(${sounds_phonetic}tango)  
exten => _499,7,Wait(1)  
exten => _499,8,Playback(${sounds_exten}T-is-not-available)  
exten => _499,9,Wait(2)  
exten => _499,10,Hangup()
```

Na telefonním čísle 499 je umístěn „echotest“. Jsou použity audio soubory ve formátu *.GSM. Cesta k danému souboru je definována prostřednictvím proměnných (\${cesta}).

```
; Jednoduche voicemenu smerujici hovory na zaklade stisknutého tlačítka  
exten => _456,1,Wait(2)  
exten => _456,2,Answer()  
exten => _456,3,DigitTimeout,10  
exten => _456,4,ResponseTimeout,20  
exten => _456,5,Playback(${sounds_exten}welcome)  
exten => _456,6,Playback(${sounds_exten}thank-you-for-calling)  
exten => _456,7,Wait(1)  
exten => _456,8,Playback(${sounds_exten}press-1)  
exten => _456,9,Playback(${sounds_exten}for-tech-support)  
exten => _456,10,Wait(1)  
exten => _456,11,Playback(${sounds_exten}press-2)  
exten => _456,12,Playback(${sounds_exten}speak-to-the-operator)  
exten => 1,1,Goto(context_support,s,1)  
exten => 2,1,Goto(context_operator,s,1)  
exten => t,1,Hangup  
  
[context_operator]  
exten => s,1,Dial(${SUPPORT})  
exten => s,2,Hangup()  
  
[context_support]  
exten => s,1,Dial(${OPERATOR})  
exten => s,2,Hangup()
```

Uvedené Voicemenu nejprve přehraje úvodní oznámení, ve kterém je uživatel přivítán a je mu nabídnuta volba „1“ pro volání Podpory a volba „2“ pro volání Operátora. Nejprve je však definován čas 20 sekund pro provedení nabídnuté volby (ResponseTimeout). Pokud je definovaný čas vyčerpán je hovor automaticky ukončen. Po provedení volby uživatelem je pomocí příslušných kontextů volání provedeno.

6.2 Session Initiation Protocol (sip.conf)

Dále uvedený konfigurační soubor je zkrácen o definování jednotlivých uživatelských čísel, jejichž definice jsou shodné s definovaným číslem „421“.

```
[general]

context=skupina           ; Default context pro prichozí volání
recordhistory=yes        ; Povolení ukládání SIP history
port=5060                 ; UDP Port (pro SIP 5060)
bindaddr=0.0.0.0         ; IP adresa pro navázání spojení (0.0.0.0 pro všechny)
srvlookup=yes            ; Povolení DNS SRV lookup na odchozí volání
tos=lowdelay              ; Nastavení QoS parametru ToS
                          ; (lowdelay,throughput,reliability,mincost,none)
disallow=all              ; Zakázání všech kodeků
allow=alaw                ; povolení vybraných kodeků
allow=gsm
allow=ulaw
musicclass=default       ; Nastavení hudby při přidržení hovoru
rtptimeout=60            ; Ukončení volání když není 60 sec RTP aktivita
                          ; (volání není ve stavu hold)
rtpholdtimeout=300      ; Ukončení volání když není 300 sec RTP aktivita
                          ; (volání ve stavu hold)
nat=yes                  ; NAT nastavení (yes, no, never, route)
```

Základní definice uživatele s číslem „421“ má následující tvar:

```
[421]
type=friend
context=skupina
language=cz
host=dynamic
username=421
secret=heslo
callerid=David <421>
```

Následující definice uživatele s číslem „428“ umožňuje použití kryptovaného zápisu hesla prostřednictvím md5. Pro generování hesla ve tvaru md5 použijeme příkaz (realm je nastaven default na „asterisk“):

```
echo -n "<user>:<realm>:<secret>" | md5sum
```

```
[428]
type = friend
context = skupina
language = cz
host = dynamic
auth = md5 ; povoleni kodovani md5
md5secret = efe5082a9b433e7e0d316bc48b911c7f ; zakodovane heslo
callerid = md5 <428>
```

Pokud chceme omezit IP adresy ze kterých je uživateli umožněno se registrovat je provedeno pomocí příkazů deny a permit.

```
[429]
deny=0.0.0.0/0.0.0.0 ; zamezeni registrace k danemu uctu ze
; vsech IP adres
permit=158.196.251.96 ; povoleni registrace k uctu z konkretni IP adresy
;permit=158.196.251.1/255.255.255.0 ; povoleni registrace k uctu ze vsech
; IP adres dane site

type=friend
context=skupina
language=cz
host=dynamic
username=429
secret=heslo
callerid=SJphone <429>
```

6.3 Inter-Asterisk Exchange (iax.conf)

Propojení mezi Asterisky jsme provedli pouze pro některá volená čísla. Na tomto popisovaném jsou definována čísla SIP/421 a SIP/424. Druhý Asterisk je na CVT Technické univerzity Ostrava a je na něm definováno číslo 420/SIP. Obě SW ústředny samozřejmě obsahují více čísel, ale vzájemná kompatibilita je řešena prozatím pouze pro tyto uvedené čísla. Prostřednictvím ústředny na CVT lze dále přistupovat také do ostatních lokalit s ústřednou Asterisk (Karviná, Opava, Orlová).

```
[general]
bindport=4569
iaxcompat=yes
delayreject=yes ; zpozdeni odpovedi na autorizaci pri pritomnosti hesla
bandwidth=high
jitterbuffer=no

disallow = all ; zakazani vsech kodeku
allow = alaw ; povoleni pouzivanych kodeku pro komunikaci
allow = ulaw
allow = gsm
allow = ilbc ; povoleno kvuli SW klientum
allow = speex ; povoleno kvuli SW klientum

tos=lowdelay
```

Pro nastavení informací o daném připojovaném serveru použijeme následující zápis. Použijeme kódovanou autorizaci prostřednictvím md5, definujeme uživatelské jméno, heslo a host IP adresu.

```
[tuo]
type=friend
auth=md5
username=asterisk
secret=testwest
host=195.113.113.24
context=virtual
trunk=yes
```

Základní definice uživatele s číslem „450“ je následující:

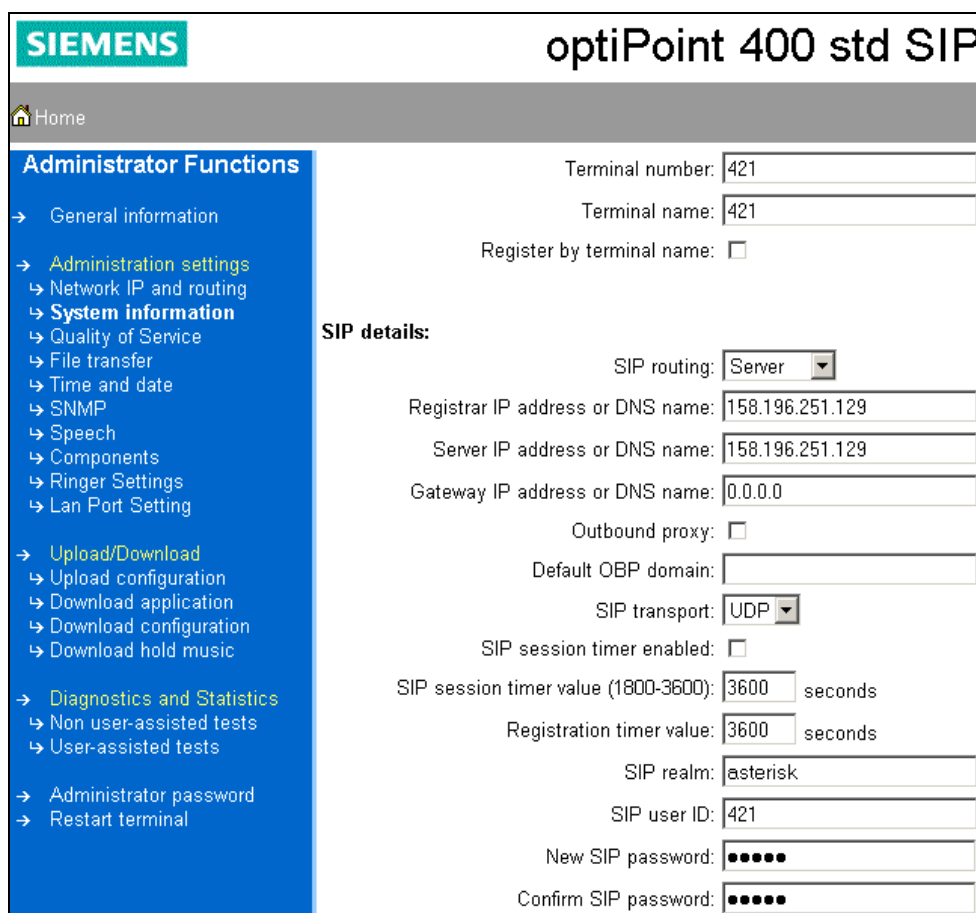
```
[450]
type=friend
context=skupina
language=cz
host=dynamic
username=450
secret=heslo
callerid=IAX1 <450>
```

Jedná se opět o základní nastavení uživatelského účtu shodné s konfigurací SIP účtů. Je tedy dále možné použít také kryptování hesla nebo omezení IP adresy, z které je možno se k danému účtu přihlásit.

7. Konfigurace SIP klientů

7.1 optiPoint 400 Standard - HW

Použili jsme telefon optiPoint 400 standard se SIP image 2.3.14. Základní konfiguraci IP adresy, síťové masky a gateway jsme provedli prostřednictvím klávesnice telefonu. Ostatní konfiguraci lze následně provést pomocí webového rozhraní. Při konfiguraci účtu prostřednictvím klávesnice jsme narazili na problém při zadávání hesla, kdy telefon po zadání a potvrzení na chvíli zatuhnul.



Obr.3: Základní konfigurace telefonu optiPoint 400 Standard

Na **Obr.3** je zobrazeno konfigurační okno „System information“, což je část konfigurace připojení telefonu k SIP serveru. Nejprve nastavíme položky „Terminal number/name“ které jsou v první fázi shodné a představují telefonní číslo uživatele. Telefon umožňuje také registraci na základě jména terminálu.

„SIP routing“ nastavíme do módu „Server“. IP adresa Asterisku je zadána v položkách „Registrar/Server IP address or DNS name“. Uživatelský účet je zadán v položce „SIP user ID“, heslo pak v dalších dvou položkách. Jedná se o jméno a heslo zadané v konfiguračním souboru Asterisku (sip.conf).

7.2 Budge Tone - 100 - HW

Použili jsme SIP telefon Grandstream Budge Tone – 100 s verzí softwaru 1.0.5.23. Základní konfiguraci IP adresy, síťové masky a gateway jsme provedli prostřednictvím klávesnice telefonu. Ostatní konfiguraci lze následně provést pomocí webového rozhraní, což velice doporučujeme. Při konfiguraci účtu prostřednictvím klávesnice se stává konfigurace vzhledem k použitému displeji nepřehlednou.

Grandstream Device Configuration		
STATUS	BASIC SETTINGS	ADVANCED SETTINGS
Admin Password:	<input type="text"/>	(purposely not displayed for security protection)
SIP Server:	<input type="text" value="158.196.251.129"/>	(e.g., sip.mycompany.com, or IP address)
Outbound Proxy:	<input type="text"/>	(e.g., proxy.myprovider.com, or IP address, if any)
SIP User ID:	<input type="text" value="498"/>	(the user part of an SIP address)
Authenticate ID:	<input type="text" value="498"/>	(can be identical to or different from SIP User ID)
Authenticate Password:	<input type="password" value="•••••"/>	(purposely not displayed for security protection)
Name:	<input type="text" value="Budgetone <498>"/>	(optional, e.g., John Doe)
Advanced Options:		
Preferred Vocoder: (in listed order)	choice 1:	<input type="text" value="current setting is 'PCMU'"/>
	choice 2:	<input type="text" value="current setting is 'PCMA'"/>
	choice 3:	<input type="text" value="current setting is 'G723'"/>
	choice 4:	<input type="text" value="current setting is 'G726-32'"/>
	choice 5:	<input type="text" value="current setting is 'G729'"/>
	choice 6:	<input type="text" value="current setting is 'iLBC'"/>
	choice 7:	<input type="text" value="current setting is 'iLBC'"/>
	choice 8:	<input type="text" value="current setting is 'iLBC'"/>

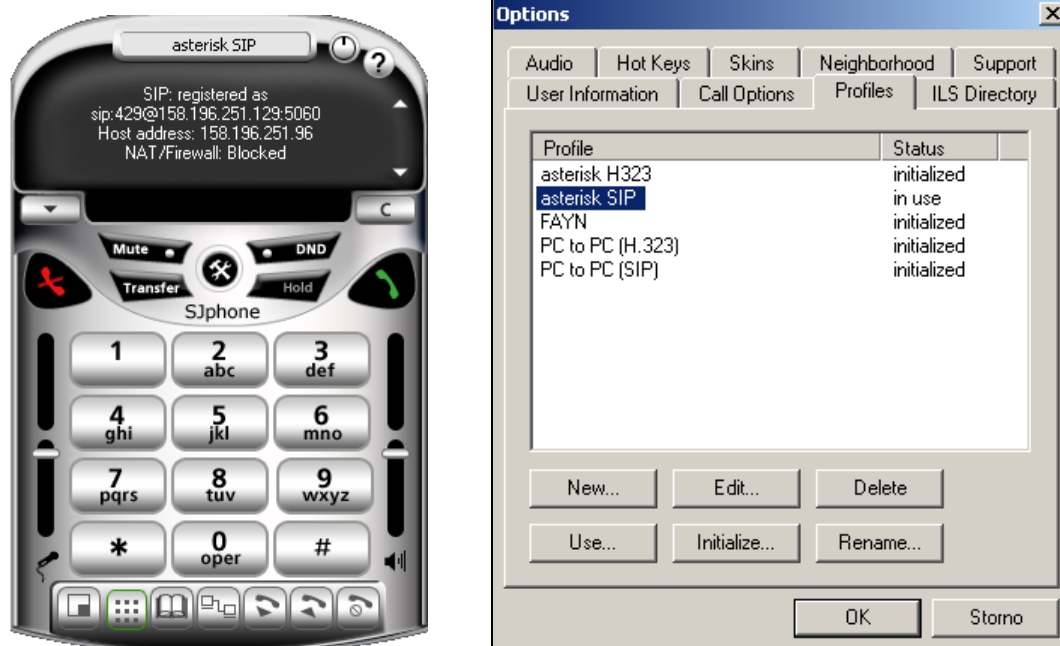
Obr.4: Základní konfigurace telefonu Budge Tone - 100

Na **Obr.4** je zobrazeno konfigurační okno „Advanced Settings“, což je část konfigurace telefonu pro připojení k SIP serveru včetně všech nastavitelných parametrů. Nastavíme adresu Asterisku „SIP Server“. Nastavení uživatelského účtu provedeme v položkách „SIP User ID / Authenticate ID / Password“. Jméno a heslo je opět dané konfigurací Asterisku (sip.conf).

Dále je nutné nastavit v položce „Advanced option“ používané kodeky. Doporučujeme nastavit kodeky na základě **Obr.4**. Důvodem je skutečnost, že pro použití v síti požadujeme hlavně podporu kodeku G.711 (položky PCMU a PCMA), dále kodek GSM, který však tento telefon nepodporuje. V konfiguračním souboru (sip.conf) jsou v základní části definované podporované kodeky (v této chvíli jsou to oba kodeky G.711 a kodek GSM), ostatní kodeky jsou zakázány z důvodu jejich omezené podpory v používaném SW a HW telefonech.

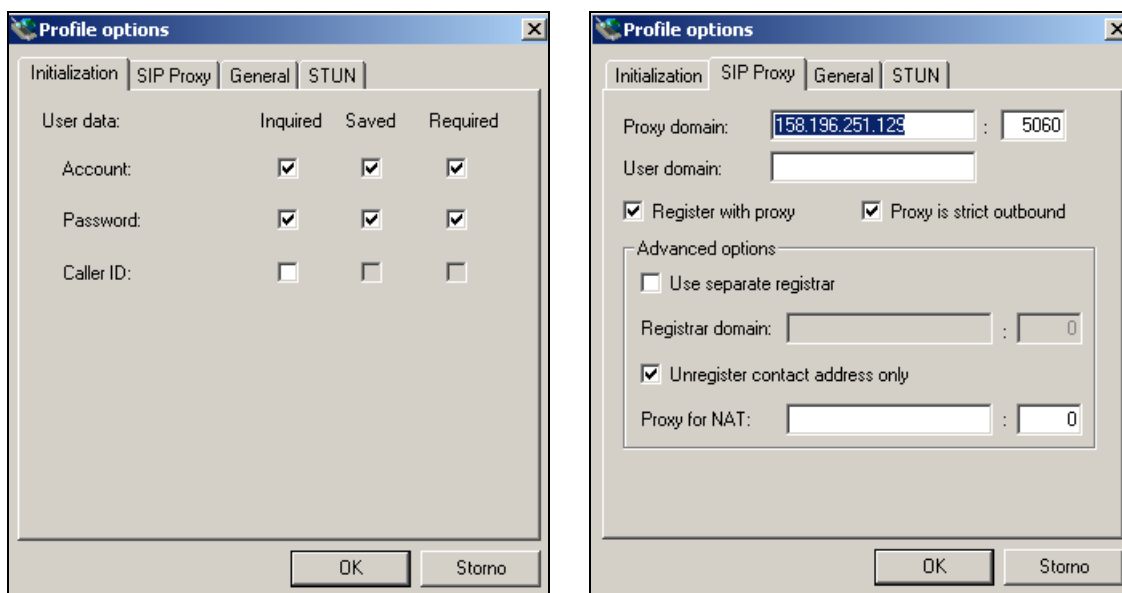
7.3 SJ Phone - SW

Použili jsme program SJ Phone Build 1.40.258 instalovaný pod OS Windows XP. Pro zaregistrování k Asterisku jsme tento program nakonfigurovali do SIP módu. Na **Obr.5** je zobrazeno základní uživatelské prostředí telefonu. Nejprve je nutné vytvořit si nový profil pomocí volby „New“, dále zadáme jméno profilu, případně souboru pod kterým bude uložen. Položkou „Profil“ zvolíme mód „Call through SIP Proxy“.



Obr.5: Prostředí SJ Phone , konfigurační prostředí

Následně konfigurujeme program způsobem, který je ukázán na Obr.6. Pokud nezaškrtneme políčko „Password – Saved“ pak bude uživatel při každém přístupu dotazován na přístupové heslo k danému účtu. Konkrétně se jedná o heslo nastavené v konfiguračním souboru Asterisk (sip.conf). Položka „Proxy domain“ je IP adresa daného Asterisku.



Obr.6: Konfigurace připojení k Asterisku

Potvrdíme provedené změny, uživatel je následně dotázán na uživatelský účet a heslo příslušející k tomuto účtu (nastaveno v souboru sip.conf). Po zadání se vrátíme do volby „Profile“. Vybereme nakonfigurovaný profil a aktivujeme jej použitím „Initialize...“. Nakonec ještě provedeme restart aplikace a softwarový telefon je zaregistrován a připraven k použití.

Použití kodeků: K výběru kodeků se uživatel dostane prostřednictvím volby „Options – Audio – Compression Settings“. Jsou dostupné kodeky GSM, G.711 a-law, G.711 μ -law a

kodeky iLBC (20ms a 30ms). Všechny tyto Asterisk podporuje. Doporučujeme použití kodeku G.711 μ -law.

7.4 Windows Messenger - SW

Nelze nastavit používané kodeky. Komunikoval bez problému s telefonem Grandstream, ale s telefonem Siemens nebylo slyšet příchozí kanál SW telefonu. Možnost registrování k Asterisku i na základě autorizace. Nutno použít službu „Komunikační služba SIP“.

Konfigurace připojení:

- Možnosti – Účty – Přihlašovací jméno (ve tvaru „jméno@Asterisk_IP_addr“)
- Možnosti – Účty – Přihlašovací jméno – Upřesnit – Název serveru nebo adresa IP (ve tvaru „Asterisk_IP_addr:5060“)
- Při přihlašování vyžaduje přihlašovací jméno ve tvaru „jméno@Asterisk_IP_addr“, uživatelské jméno a heslo.

Nutno následně vytvořit seznam kontaktů nebo volat prostřednictvím volby Zahájit hlasovou konverzaci – Jiné – zadat volbu ve tvaru „číslo@Asterisk_IP_addr“.

Použití tohoto SW telefonu nedoporučujeme !!!

7.5 X-lite - SW

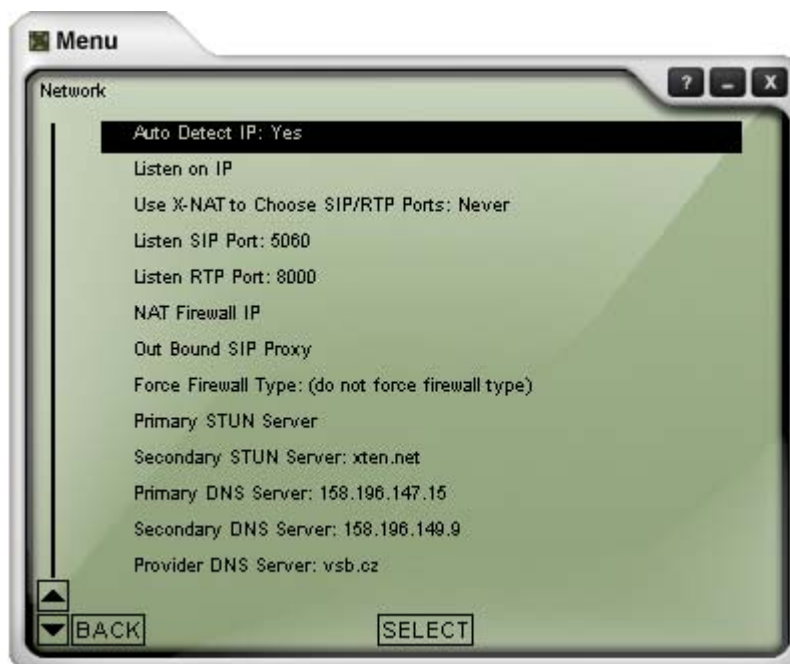
Použili jsme program X-Lite v.2.0 Build 1103m pod OS Windows XP (SIP softphone). Pro zaregistrování k Asterisku jsme tento program nakonfigurovali do SIP módu. Na **Obr.7** je zobrazeno základní uživatelské prostředí telefonu. Všechna důležitá nastavení se prováděla v položce „System Settings“.



Obr.7: Základní prostředí X-lite

V nabídce „System Settings“ jsou důležité položky „Network“ a „SIP Proxy“. V nabídce „Network“ (**Obr. 8**) se nastavují informace o DNS serveru. V nabídce „SIP Proxy“

(Obr. 9) nastavíme uživatelské jméno a heslo (definované v Asterisku), doménu a SIP Proxy (IP adresy Asterisku).



Obr. 8: Síťové nastavení



Obr. 9: SIP Proxy nastavení

Nakonec provedeme restart aplikace a softwarový telefon je zaregistrován a připraven k použití.

Použití kodeků: K výběru kodeků se uživatel dostane prostřednictvím volby „Advance System Settings – Codec Settings“. Dostupné jsou kodeky GSM, G.711 a-law, G.711 μ -law, iLBC a Speex. Všechny tyto Asterisk podporuje. Povolené kodeky jsou zobrazeny na displeji telefonu a při každém volání je zvýrazněn ten, který je právě používán.

8. Konfigurace IAX klientů

Pro správnou funkci SW klientů je potřeba mít korektně nakonfigurovaný soubor `iax.conf` (viz. Kapitola 6.3). Kromě klasických kodeků G.711 a-law, G.711 μ -law a GSM. Je možné také povolit kodeky iLBC a Speex, které softwaroví klienti podporují. Použití těchto kodeků však nedoporučujeme. Asterisk tyto hovory sice spojí, ale generuje chybovou hlášku „Operating with different codecs, can't native bridge...“, tedy nepovažuje spojení za „přirozené“.

8.1 Ethernet Phone AT-320 - HW

IP telefon AT-320 je levným HW řešením pro VoIP. Jeho výhodou je, že podporuje protokoly H.323, SIP, MGCP a nově také protokol IAX2.

Nejprve je nutno prostřednictvím klávesnice telefonu konfigurovat IP adresu, síťovou masku a gateway. Poté je telefon restartován. Následnou konfiguraci doporučujeme provést prostřednictvím webového rozhraní.

K telefonu se přihlásíme pomocí IP adresy a default hesla „1234“. Konfigurační menu je zobrazeno na **Obr.10**.

Network Settings			
iptype	static	ppp id	
local ip	158.196.251.205	subnet mask	255.255.255.0
dns	0.0.0.0	dns2	0.0.0.0
ppp pin		router ip	158.196.251.1
		mac	00-09-45-63-8e-6f
Audio Settings			
codec1	g711u	codec2	gsm
codec4	null	codec5	null
codec3	g711a	codec6	null
vad	<input type="checkbox"/>	agc	<input type="checkbox"/>
audio frames	2	g.723.1 high rate	<input checked="" type="checkbox"/>
jitter size	0		
Phone Settings			
use dialplan	disable	dial number	
idd code	86	ddd code	10
idd prefix	00	ddd prefix	0
inner line	disable	inner line prefix	0
call waiting	<input type="checkbox"/>	forward number	82378009
fwd poweroff	<input type="checkbox"/>	fwd noanswer	<input type="checkbox"/>
fwd always	<input type="checkbox"/>	fwd busy	<input type="checkbox"/>
answer	30	use digitmap	<input type="checkbox"/>
handset in(0-15)	7	handset out(0-31)	20
ring type	dtmf	speaker out(0-31)	20
speaker in(0-15)	4		
IAX2 Protocol Settings			
use service	<input checked="" type="checkbox"/>	service addr	158.196.251.129
register ttl	120	phone number	450
account	450	pin	heslo
local port	5036	dtmf	outband signal
super password	12345678	debug	no check

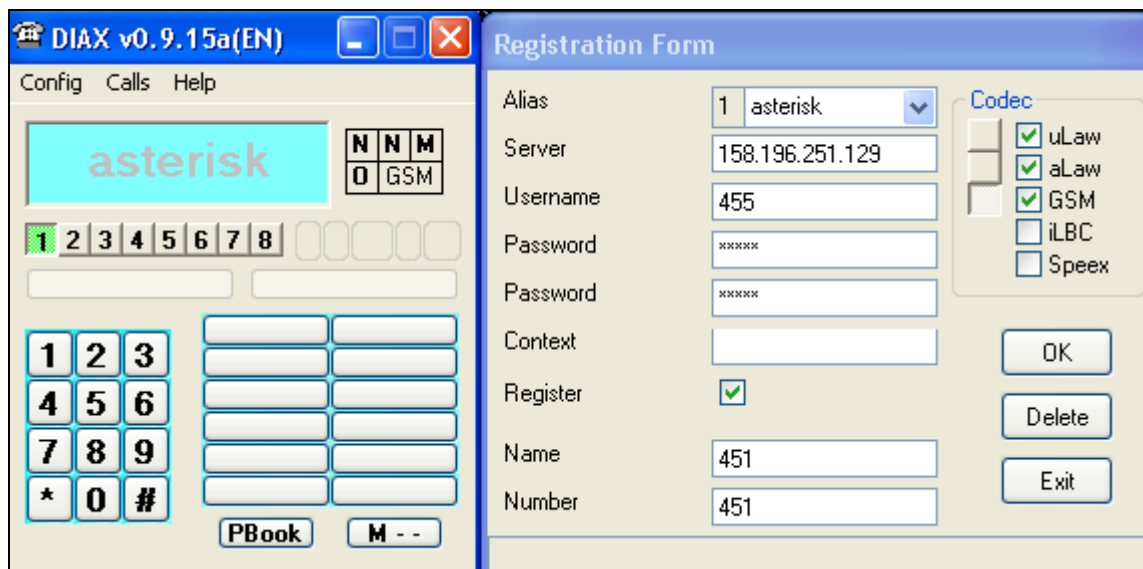
Obr.10: Konfigurační menu telefonu AT-320

Pro zaregistrování se k Asterisku provedeme následující nastavení. Jedná se konkrétně o položky „IAX2 Protocol Settings“.

V telefonu je nahrán image IAX v.1.42.028. „Service addr“ je IP adresou Asterisku. „Phone number“ je telefonní číslo daného účtu. Uživatelský účet / heslo pod kterým se telefon registruje k Asterisku je zadán ve volbě „account / pin“. „Local port“ nastavíme na hodnotu 5036.

8.2 DIAX - SW

Použili jsme program DIAX v.0.9.15a pod OS Windows XP (IAX softphone). Klient podporuje registraci s ústřednou Asterisk, registraci s více servery, protokol IAX2, nezávislá konfigurace kodeků pro jednotlivé servery, umožňuje také připojení USB telefonu, atd. Nastaví registračních údajů pro připojení k ústředně Asterisk je zobrazeno na **Obr.11** (menu „Config – Registration“).



Obr.11: Základní prostředí DIAX, Registrace k ústředně Asterisk

Položka „Alias“ představuje používaný profil připojení k serveru, je možno konfigurovat 8 různých profilů s různými parametry (včetně audio kodeků). Username a Password je shodné s nakonfigurovaným účtem v ústředně. Označením položky Register donutíme SW telefon aby se registroval na základě jména a hesla. Adresu serveru lze v případě potřeby zadat včetně příslušného portu.

Použití kodeků: K výběru kodeků se uživatel dostane prostřednictvím volby „Config - Registration“. Jsou dostupné kodeky GSM, G.711 a-law, G.711 μ -law, kodek iLBC a kodek Speex. Všechny tyto Asterisk podporuje. Doporučujeme použití kodeku G.711 a-law nebo GSM, při použití kodeku G.711 μ -law nebyl hovor několik sekund slyšet. Při výběru více kodeků lze jeden z nich vybrat jako preferovaný. Lze také použít kodeky iLBC a Speex, u těchto však není zaručena kompatibilita s klienty HW. Několikrát se nám také stalo že hovor byl spojen se zpožděním a na straně HW telefonu jej nebylo možné vyzvednout (je to otázkou jestli to byla chyba SW telefonu, HW telefonu nebo Asterisku).

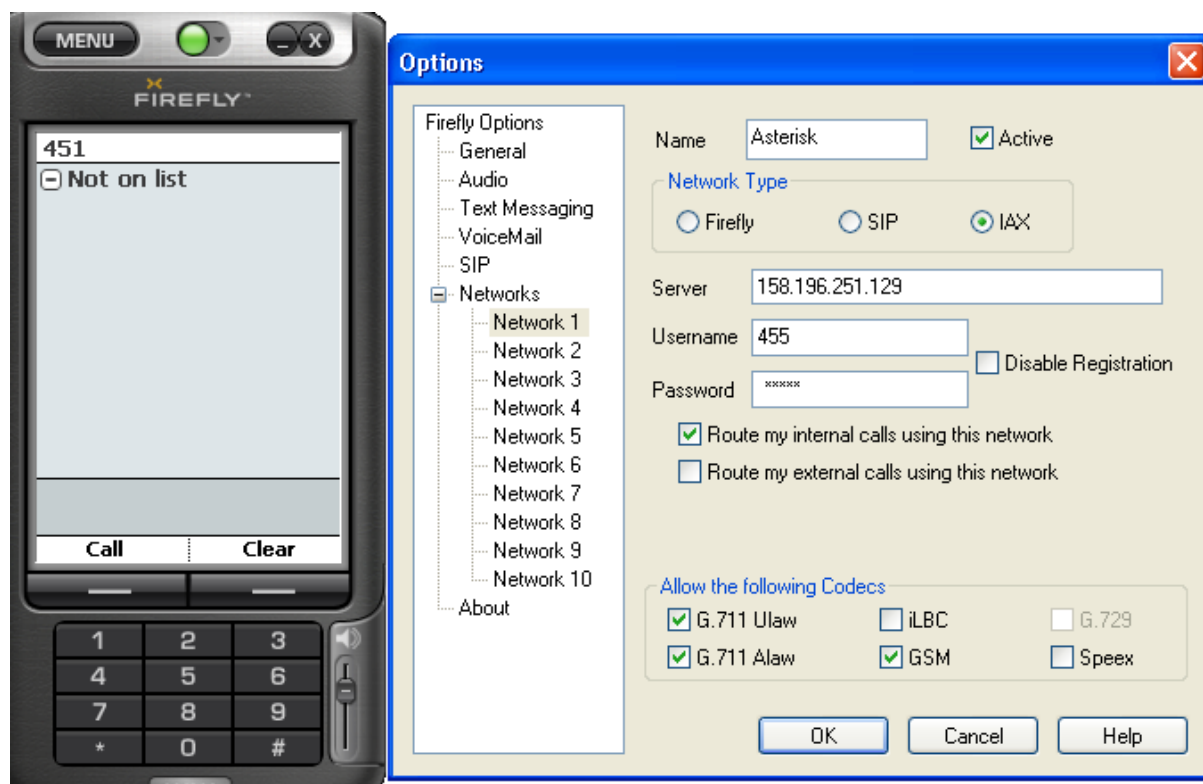
8.3 FireFly - SW

Použili jsme program FireFly v.1.9.9 pod OS Windows XP (IAX, SIP softphone). Klient podporuje registraci s ústřednou Asterisk, protokoly IAX, SIP, nezávislá konfigurace až 10 různých serverů, možnost registrace do sítě FireFly, vytvoření účtu a volání do komerčních sítí po celém světě.

Při instalaci se objeví uvítací okno „Welcome to FireFly“. Zde zvolte možnost „I wish to connect to a third party network“, což umožní nakonfigurovat program pro použití v IAX nebo SIP sítích. V dalším okně „Third network setup“ nastavíte konfigurační údaje pro

registraci k ústředně Asterisk. Server, Username a password jsou opět konfigurované parametry v ústředně.

Nastavené parametry si můžeme ověřit a dokonfigurovat prostřednictvím volby „Options“ zobrazené na **Obr.12** (menu „Menu – Option – Network – Network 1“).



Obr.12: Základní prostředí FireFly, nastavení uživatelského účtu

Volbu „Disable Registration“ ponecháme neaktivní, telefon se bude tedy automaticky registrovat k ústředně. Volbou „Activate“ v pravém horním rohu určíme, že právě tento daný profil bude používán.

Použití kodeků: K výběru kodeků se uživatel dostane prostřednictvím shodného postupu jako ke konfiguraci připojení. Jsou dostupné opět kodeky GSM, G.711 a-law, G.711 μ -law, kodek iLBC a kodek Speex. Všechny tyto Asterisk podporuje. Doporučujeme použití některého z kodeků G.711 a-law, GSM nebo G.711 μ -law. Lze také použít kodeky iLBC a Speex, u těchto však není zaručena kompatibilita s klienty HW. Na Internetu je dostupná také knihovna s kodekem G.729, vzhledem ke komerční povaze tohoto kodeku však tento konkrétní popisovaný Asterisk nemá jeho podporu.

9. Odkazy na Internet

9.1 SW klienti

- | | | | |
|-----|-------------------|------------|--|
| [1] | SJ-Phone | SW telefon | SIP, H.323
http://www.sjlabs.com |
| [2] | Windows Messenger | SW telefon | SIP
http://www.microsoft.com |
| [3] | X-lite | SW telefon | SIP
http://www.xten.com/ |
| [4] | DIAX | SW telefon | IAX
http://www.laser.com/dante/diax/diax.html |
| [5] | FireFly | SW telefon | SIP, IAX
http://www.freshtel.net/firefly/ |