

VoIP ústředna s protokolem SIP

VoIP switchboard with SIP protocol

Bc. Tomáš Iglo

Diplomová práce
2009



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav aplikované informatiky

akademický rok: 2008/2009

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Tomáš IGLO**

Studijní program: **N 3902 Inženýrská informatika**

Studijní obor: **Informační technologie**

Téma práce: **VOIP ústředna s protokolem SIP**

Zásady pro vypracování:

Nakonfigurujte VOIP ústřednu s protokolem SIP tak, aby poskytovala:

1. **Funkce ENUM.**
2. **Volání do externích pevných/mobilních sítí.**
3. **www rozhraní pro seberegistraci nových uživatelů.**

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. Informace o telefonním systému IP PBX/VOIP/SIP [online]. [cit. 2008-1-23].
Dostupný z WWW: <http://www.3cx.cz/voip-sip/ip-pbx-faq.php>.
2. SIP [online]. [cit. 2008-1-23]. Dostupný z WWW:
<http://www.voip-info.org/wiki-SIP>.
3. Pomoc, návody a často kladené dotazy o SIP IP telefonii [online]. [cit. 2008-1-22].
Dostupný z WWW: <https://sip.osu.cz/hlp/index.php>.
4. ENUM – Nové horizonty [online]. [cit. 2008-1-22]. Dostupný z WWW:
http://www.iptw.cz/files/ENUM-Nove_horizonty.pdf.
5. ENUM – systém pro veřejnou komunikaci přes protokol SIP [online]. [cit. 2008-1-20]. Dostupný z WWW:
<http://www.computerworld.cz/cw.nsf/id/7CA63B75AFF0219AC12571E2003CE1A0?OpenDocum>

Vedoucí diplomové práce:

Ing. Tomáš Dulík

Ústav aplikované informatiky

Datum zadání diplomové práce:

20. února 2009

Termín odevzdání diplomové práce:

27. května 2009

Ve Zlíně dne 13. února 2009

prof. Ing. Vladimír Vašek, CSc.
děkan



doc. Ing. Ivan Zelinka, Ph.D.
ředitel ústavu

ABSTRAKT

Cílem této práce je nakonfigurovat pobočkovou ústřednu (PBX), která poskytuje pomocí programů Asterisk a OpenSER kompletní řešení pro volání do všech sítí (PSTN, VoIP, mobilní telefony) a to díky řízení Asteriskem a jeho pravidel. Díky bráně SPA3102 bude možné volat i z pevné sítě na PBX. Samozřejmě bude volání v obou směrech. Tato PBX by měla vystačit pro běh menší až střední firmy a to především díky programu OpenSER, který je vhodný pro větší zátěž volání v oblasti VoIP SIP protokolu.

Klíčová slova: VoIP, OpenSER, Asterisk, SIP

ABSTRACT

The aim of this work is to configure the switchboard (PBX), which provides a complete solution for calls to all networks (PSTN, VoIP and mobile phones) thanks to the Asterisk and its call routing rules. Gateway SPA3102 is used for interfacing the fixed network to the Asterisk PBX. The PBX should can be used in small or medium organizations, mainly thanks to the OpenSER, which is suitable for larger call loads.

Keywords: VoIP, OpenSER, Asterisk, SIP

Tímto bych chtěl poděkovat panu Petru Hruškovi z NIC.CZ, který mi pomohl s některými problémy, které nastaly během této diplomové práce, dále bych chtěl poděkovat panu Ing. Tomáši Dulíkovi za jeho trpělivost, rady a návrhy, co by měla tato práce vše poskytovat.

Prohlašuji, že

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

Prohlašuji,

že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval.
V případě publikace výsledků budu uveden jako spoluautor.

Ve Zlíně

.....
Podpis diplomanta

OBSAH

ABSTRACT	4
ÚVOD	9
I TEORETICKÁ ČÁST	11
1 TEORIE VOIP VOLÁNÍ	12
1.1 SIP.....	13
1.1.1 Adresace v SIP protokolu.....	14
1.1.2 SIP server.....	15
1.1.3 Stavové a bezstavové SIP Proxy.....	15
1.2 TYPICKÝ SCÉNÁŘ SPOJENÍ.....	16
1.3 REGISTRACE A PŘESMĚROVÁNÍ.....	17
1.4 METODY A ODPOVĚDI.....	18
1.4.1 Žádost.....	18
1.4.2 Specifikace žádostí (metod).....	19
1.5 ODPOVĚDI.....	20
1.6 REGISTRACE NA SIP SERVERU.....	24
2 PROGRAMY PRO KOMUNIKACI	25
2.1 OPENSER.....	25
2.1.1 Historie projektu.....	25
2.1.2 Co je to OpenSER.....	26
2.2 ASTERISK.....	26
2.2.1 Využití Asterisku.....	27
2.2.2 Podpora kodeků.....	27
2.3 POROVNÁNÍ OPENSERU A ASTERISKU.....	28
2.3.1 Architektura.....	28
2.3.2 NAT Traversal.....	29
2.3.3 Load Balancing.....	29
2.3.4 Nízkoúrovňové přístupy k SIP hlavičkám a transakcím.....	29
2.3.5 Integrace s Radius, Diameter a LDAP.....	29
2.3.6 Služby médií.....	29
2.3.7 Konektivita do PSTN.....	29
2.3.8 Závěr porovnání.....	30
3 ENUM	31
3.1 CO JE ENUM?.....	31
3.2 ROZDÍL MEZI VOIP A ENUM.....	31
3.2.1 Běžný hovor přes VoIP.....	31
3.2.2 Hovor díky zaregistrovanému číslu pomocí ENUM.....	31
3.3 JAK FUNGUJE VOLÁNÍ S ENUM.....	32
II PRAKTICKÁ ČÁST	33
4 KONFIGURACE SIP SERVERU	34

4.1	INSTALACE A KONFIGURACE OPENSERU.....	35
4.1.1	Instalace.....	35
4.1.2	Konfigurace.....	35
4.1.3	Instalace, nastavení MySQL + přidání uživatele do OpenSERu.....	37
4.1.4	Úprava openser.cfg.....	38
4.1.5	Instalace, nastavení programů X-Lite, WengoPhone.....	44
5	REALTIME INTEGRACE PROGRAMŮ ASTERISK A OPENSER.....	49
5.1	INSTALACE, KONFIGURACE ASTERISKU.....	49
5.1.1	Konfigurace hlavních souborů Asterisku.....	51
5.2	UKONČENÍ DOSAVADNÍ MYŠLENKY DIPLOMOVÉ PRÁCE.....	53
6	PBX ASTERISK.....	54
6.1	ZÁKLADNÍ OVLÁDÁNÍ ASTERISKU.....	54
6.2	SPA3102 BRÁNA.....	55
6.2.1	Nastavení SPA - Line1 tab.....	55
6.2.2	České oznamovací tóny.....	58
6.2.3	PSTN tab.....	59
6.2.4	User 1 tab.....	61
6.3	KONFIGURAČNÍ SOUBOR SIP.CONF.....	62
6.4	NÁVRH DIAL PLÁNU.....	64
6.4.1	PSTN síť – PSTN číslo O2 (příchozí a odchozí hovory).....	64
6.4.2	VoIP operátor (příchozí a odchozí hovory).....	65
6.4.3	Volání v rámci lokální sítě (příchozí i odchozí hovory).....	65
7	ENUM.....	68
8	WWW ROZHRANÍ PRO SEBEREGISTRACI NOVÝCH UŽIVATELŮ.....	70
	ZÁVĚR.....	71
	ZÁVĚR V ANGLIČTINĚ.....	73
	SEZNAM POUŽITÉ LITERATURY.....	76
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	79
	SEZNAM OBRÁZKŮ.....	80
	SEZNAM TABULEK.....	81
	SEZNAM PŘÍLOH.....	81

ÚVOD

Cílem práce je konfigurace serveru (ústředny) pro VoIP volání pro menší firmu, které jde zejména o ušetření a minimalizaci nákladů za volání do telefonních sítí. Práce by měla obsahovat samotné použití a návod, jak nakonfigurovat VoIP server a použít jej pro dané účely.

Konfigurace serveru bude probíhat tím způsobem, že po zaregistrování u jednotlivých VoIP providerů získáme např. levnější volání do 12:00 hod. u dalšího providera volání od 12:00-16:00 hod. a u dalšího od 16:00-24:00 hod. Popř. na některá čísla bude výhodnější volání u jednoho providera a na jiné typy čísel bude zase výhodný jiný provider. Díky tomu bude možné vybrat si providera, který je v rámci volaného čísla nejlevnější, kvůli ušetření nákladů na volání. Tento způsob by spíše zahrnoval volání na pevné síť nebo do zahraničí pomocí VoIP providerů, kdy je volání levnější než na mobilní síť, kde jsou vysoké propojovací poplatky operátorům.

Větší problém jsou mobilní síť, kdy je volání pomocí VoIP celkem drahé, a tak se tímto zvyšují náklady za volání. Tento problém by se dal řešit pomocí GSM brány (např. EasyGate), kdy by bylo možné tyto náklady velmi lehce minimalizovat. A to tím způsobem, že by do tohoto zařízení byla vložena SIM karta, která umožňuje výhodné volání nebo volání zdarma, vybraným jedincům. Např. T-Mobile nabízí tuto možnost v tarifním zvýhodnění „Rodina“, kdy je možné 4 členům volat v síti T-Mobile zcela zdarma a jakkoli dlouho. Tímto by se dala vyřešit situace nejvýhodnějšího volání mezi nejdůležitějšími členy firmy. Brána EasyGate by komunikovala se serverem pomocí programu Asterisk, který umožňuje jednodušší konfiguraci této brány i komunikaci vzájemně mezi serverem a EasyGate bránou, kde by bylo velmi obtížné nebo dokonce nemožné nakonfigurovat tento modul do programu OpenSER. Bohužel z časových i finančních důvodů nešlo pořídit pro mou diplomovou práci obě zařízení, které by řešily jak nejlevnější volání do mobilních sítí pomocí GSM brány EasyGate, tak i volání na číslo zřízené u PSTN poskytovatele s využitím Linksys SPA3102 bráně. Proto byla poskytnuta pouze Linksys SPA3102 brána, která umožňuje připojení PSTN (pevná síť) a běžného analogového telefonu k programu Asterisk, který s ní dokáže komunikovat. A tak je umožněno v této diplomové práci se zabývat hovory přes SPA3102 a PSTN síť a poté volání přes VoIP providery a samozřejmě standardní VoIP volání pomocí protokolu SIP.

Další důležitou vlastností bude přesměřování telefonních hovorů po lokální síti nebo mezi jednotlivými sítěmi. Toto vše bude obstarávat Asterisk, který bude řídit všechny

účty a pomocí tzv. dialplanu se bude rozhodovat o řízení odchozích a příchozích volání. Pro zapojení Asterisku do lokální sítě nám může pomoci NAT Helper, který je implementován v programu OpenSER a který nám poskytne překlad adres do další podsítě. Další vlastností, kterou by mělo volání poskytovat, je tzv. ENUM. Jde o obousměrný překlad adres na zaregistrovaná VoIP telefonní čísla, který umožňuje bezplatné volání na volané číslo.

I. TEORETICKÁ ČÁST

1 TEORIE VOIP VOLÁNÍ

Za počátek IP telefonie je považován únor 1995, kdy izraelská společnost Vocaltec přišla s komerčním produktem Internet Phone. Nové zpracování hlasu a použití kodeků s lineární predikcí, zajistily redukci hovorového pásma a zajistily tak možnost použití v IP telefonii. Přední použité kodeky byly dle ITU – T G.729 a G.723.1, avšak povinné bylo dodržet použití ITU – T G.711. Využití IP telefonie pro telefonování v rámci internetu tak snižuje náklady a díky rozvoji IP telefonie se snižují provozní režie a zvyšuje se konkurence schopnost služeb, což vede k neustálému rozvoji a snižování cen. Z toho vyplývají hlavní otázky týkající se bezpečnosti (zneužití VoIP systémů, spam, zjištění utajení hovoru, apod.)

Výhody, lze vidět v dvou hlavních oblastech:

- efektivnější řešení, využití jedné přenosové infrastruktury,
- nové možnosti, větší mobilita, ENUM (mapování telefonních čísel a URI adres přes systém DNS), snadnější integrace aplikací (např. tel. seznam v telefonu na bázi LDAP), nové služby typu Instant Messaging, prezence (zobrazení stavu konkrétního účtu – odhlášen, přihlášen, na obědě, atd...).

Nevýhody z používání IP telefonie by se dali shrnout do dvou oblastí:

- IP telefonie přináší ve srovnání s PSTN snížení spolehlivosti a dostupnosti služby (uvádí se nižší o 0,5 %),
- bezpečnostní rizika zmíněná výše.

Voice over IP (VoIP) byla určena pro volání pouze skrz Internet, pro propojování jen samotných IP telefonů, více méně v podobě softwarových telefonů instalovaných na PC, z toho vyplývá, že propojení s okolními sítěmi nebylo vyřešeno. Nyní je již situace zcela opačná a IP telefonie umožňuje plnohodnotnou náhradu za jakýkoli telefonní přístroj.

Po síti s IP se hlas přenáší díky RTP paketům (Real Time Protocol), které využívají na transportní vrstvě UDP. Formát tohoto paketu je dán díky IETF z roku 1996 a má označení RFC 1889/18890 pro RTP/RTCP, kde RTP řeší přenos hlasové informace a RTCP zajišťuje kontrolní mechanismus při doporučení RTP. Vznikly i novější implementace, které se nyní používají jako např. RFC3550 z roku 2003 nebo ještě novější SRTP (Secure RTP) podle RFC3711 z roku 2004. Během přenosu RTP dostáváme informace o počtech ztracených paketů a vzniklém zpoždění pomocí kontrolního protokolu RTPC. Zajímavým rozšířením RTCP je RTCP XR (Extended Reports), ten definuje soubor

metrik pro hodnocení VoIP kvality volání. RTCP XR umožňuje zobrazení hodnot MOS nebo také R-faktoru na koncových zařízeních. Převážnou výhodou tohoto protokolu je sledování kvalitativních parametrů a jejich následné vyhodnocení kvality pro konkrétní destinace. Což někdy vede ke změně směrování volání v síti. (použití např. v síti AARNET).[6]

Dalším protokolem v IP telefonii je SIP protokol. Bude hlavní částí této diplomové práce. Poprvé byl SIP uvolněn v roce 1999 a dnes je dostupný ve dvou verzích RFC, k SIPu se váže ale dalších zhruba sedmdesát RFC, které rozšiřují jeho možnosti.

1.1 SIP

SIP (Session Initiation Protocol - protokol pro inicializaci relací) byl vyvíjen od roku 1996 pracovní skupinou MMUSIC (Multiparty Multimedia Session Control) v rámci IETF (Internet Engineering Task Force). Byl předložen ve formě navrhovaného standardu RFC 2543 v roce 1996. Tohoto roku vznikla skupina nazvaná SIP, která začala vyvíjet hlavní jádro tohoto signalizačního protokolu, který pracuje na aplikační vrstvě. Byl navržen, aby se dal snadno implementovat, rozšířit a aby byl dostatečně flexibilní. Nejvíce specifické doporučení pro tento protokol je RFC3261, které obsahuje samotné jádro protokolu. Je využit pro sestavení, modifikaci a ukončení spojení mezi dvěma či více účastníky.

SIP protokol byl navrhnut, aby vyhovoval modelu Internetu. Je to end-to-end protokol, a to znamená, že koncová zařízení mají uloženu veškerou logiku a znají i jednotlivé stavy komunikace, tímto je zvýšena odolnost komunikace před vzniklými chybami. SIP protokol a jeho řešení díky end-to-end konceptu je významně odlišný oproti klasické PSTN (Public Switched Telephone Network), kde je jeho logika uložena v síti a díky tomu mohou být koncová zařízení jednodušší. Hlavní devizou protokolu SIP je využití end-to-end konceptu, který přináší vyšší výkonnost a umožňuje nasazení nových služeb, které nemohou být do PSTN nasazeny. Přes protokol SIP se komunikace provádí výměnou dvou typů zpráv, a to požadavků a odpovědí. Klient i server jsou logickou částí jednoho celku.

SIP je podobný HTTP protokolu, který má formát hlaviček díky RFC822. HTTP je nejpoužívanější protokol v Internetu a ve své podstatě se jedná také o signalizační protokol, protože UA (user agents) jej používají pro sdělení serveru o jaký dokument se zajímají. SIP se využívá pro přenos parametrů a relace. Samotný popis je zakódován dovnitř dokumentu používajících SDP. Oba protokoly tak obsahují stejné kódování hlaviček zpráv z RFC822.

Tato implementace formátu zaručuje velkou robustnost protokolu.[6]

1.1.1 Adresace v SIP protokolu

SIP protokol je vázán k doméně, což respektuje adresace. Uživatel existuje v konkrétní doméně, kterou obsluhuje SIP server. SIP entity jsou identifikovány použitím SIP URI (Uniform Resource Identifier).

SIP URI má tento formát:

;sip:user@host:porturi-parameters

Jak je vidět, SIP URI se skládá z části *user* a z části *host*, obě části jsou oddělené znakem *@*. SIP URI je podobná e-mailové adrese, a proto je zde možnost využívat obou služeb na jedné adrese pro snazší zapamatování.

Část *user* ukazuje na uživatele, který je v doméně v části *host* (za *@*). Zde je možnost zadat, jak jméno tak i IP adresu serveru. Pokud není uveden port, tak se předpokládá použití všeobecně známého portu 5060. Parametry mohou nést další volitelné informace. Doménová část URI je adresována s využitím DNS, což dává adresaci vysokou flexibilitu.[6]

V tabulce 1 jsou uvedeny příklady URI a jejich doporučení.

Tabulka 1: URI a odkazy na doporučení [6]

URI	Použití adresy	Doporučení
sip: nebo sips:	SIP a Secure SIP adresa	RFC 3261
tel:	Telefonní čísla	RFC 3999
im:	Instant Message	RFC 3861
http:	Web	RFC 2616

1.1.2 SIP server

Infrastrukturu lze vytvořit díky SIP protokolu jako proxy servery, a tak mohou UA (user agents) odesílat zprávy přímo na proxy server. Proxy servery jsou díky zajišťování směrování žádostí o spojení, autentizaci, účtování a dalších funkcí, důležitou součástí v SIP infrastruktuře sítě. Jejich hlavní úlohou je směrování žádostí o sestavení spojení co nejbliž k volanému. Proxy server prohledává i několik dalších serverů, dokud nenajde server s aktuálním odkazem na volaného. Čili proxy server přesměrovává žádosti o spojení přímo k volanému a volaný se sám rozhodne zda přijme daný hovor nebo odmítne. Rozlišujeme dva druhy SIP proxy serverů:

- stateless (bezstavový),
- stateful (a s informací o stavech).

1.1.3 Stavové a bezstavové SIP Proxy

Bezstavové (stateless) servery jsou jednodušší, protože přeposílají zprávy nezávisle na jejich vzájemné vazbě, tyto servery především neumí kontrolovat výměnu, což může někdy vést nekorektním stavům, které musí ošetřit koncové zařízení. Díky tomu jsou rychlejší než stateful servery, a proto jsou využívány ke snížení zátěže pro překlad zpráv a jejich směrování. Hlavní nevýhodou je, že nedokáží zachytit opakování zpráv a provádět díky nim přesnější směrování. Ze signalizace jde jasně poznat, že se jedná o stateless server, protože dané zprávy jenom přeposílá.

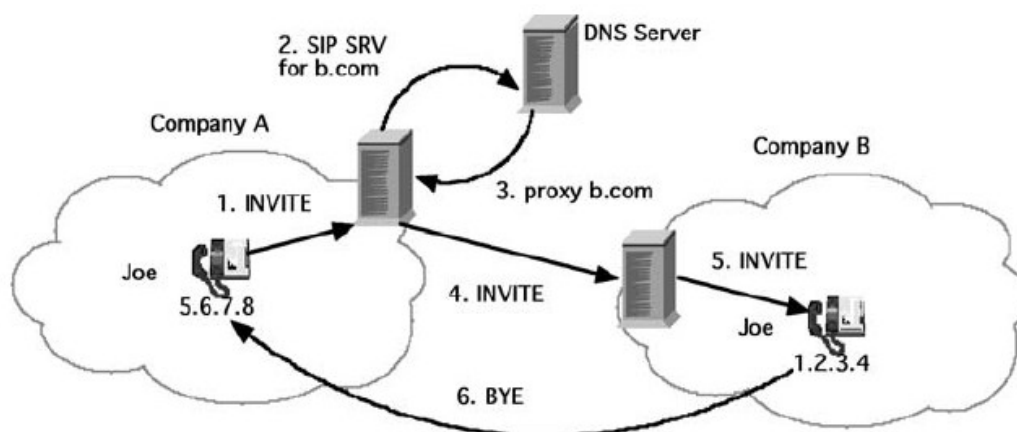
Stavové (stateful) SIP proxy servery jsou mnohem komplexnější díky informacím o stavech. Když přijde na server požadavek, server si jej ponechá dokud nedojde k ukončení transakce. Např. zpráva INVITE může trvat poměrně dlouho, dokud si jej volaný nevyzvedne nebo se neukončí volání, a proto je jejich výkon limitován.

Díky tomu, že tento server dokáže do transakcí přiřazovat SIP zprávy, mu umožňuje např. provádět větvení, na přijetí zprávy a může být odesláno více zpráv. Může dokonce zachytit opakování zpráv, díky transakci lze zjistit, zda-li byla zpráva už přijata. Stateful proxy server může využít komplikovanějších metod nalezení uživatele, např. přesměrovat pro nečinnost na jiný telefon. Toto stateless server neumožní, protože nepozná, zda-li byla zpráva přijata. Dle informativních odpovědí proxy nejdříve odpoví a až poté ji přepoše dál. Můžeme je dále dělit takto:

- Transaction stateful proxy, což je transakční proxy udržující stav transakce, tzn. od přijetí požadavku až po odeslání konečné odpovědi.[6]
- Call stateful proxy, což je dialogová proxy udržující stav celého dialogu, tzn. od přijetí první žádosti INVITE až po ukončovací BYE.[6]

1.2 Typický scénář spojení

Typická konfigurace vypadá asi následovně, každá firma má svůj vlastní SIP server, který využívají všichni UA využívající převážně jednu doménu. Lze však používat i tzv. multidoménovou SIP proxy. Ta je sice umístěna v konkrétní doméně, ale pomocí realm pole dokáže určit, do jaké domény uživatel patří. Předpokládejme, že jsou dvě firmy A a B a každá z nich má svůj proxy server ve své doméně. Obrázek ukazuje jak zaměstnanec Joe ve firmě A inicializuje spojení se zaměstnancem Bobem z firmy B.



Obr. 1: Inicializace spojení Joe – Bob [6]

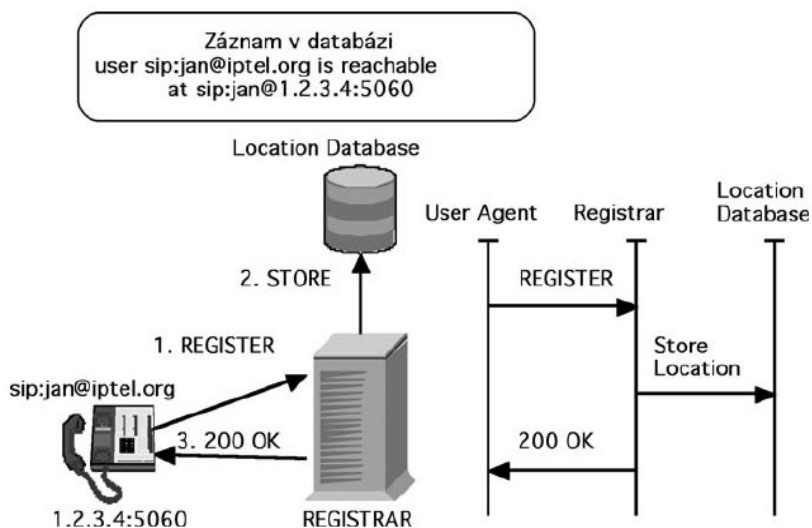
Uživatel Joe volá Boba a použije adresu *sip:bo sip:bob@b.com*. UA neví, kam má poslat žádost o sestavení spojení, ale je nakonfigurován tak, že všechny odchozí provoz (outbound traffic) posílá na SIP proxy server své firmy s adresou *proxy.a.com*. Proxy server zjistí, že uživatel *sip:bob@b.com* je v jiné firmě, a tak vyhledá odpovídající SIP proxy server, kam pošle žádost. Odpovídajícím serverem je *proxy.b.com* a je zadán staticky v proxy serveru firmy A anebo bude proxy vyhledán pomocí záznamu DNS SRV. Žádost tedy dorazí na *proxy.b.com*. Proxy ví, že Bob je aktuálně ve své kanceláři a dosažitelný na telefonu na svém stole, jež má IP adresu 1.2.3.4, takže proxy na ní posílá žádost.

Bylo zmíněno, že SIP proxy na *proxy.b.com* zná současnou polohu Boba, ale nebylo řečeno, jak proxy může lokalizovat uživatele. Bobův UA (SIP telefon) musí být registrován na tzv. *registrar serveru*. Registrar server je speciální část SIP serveru, která přijímá od uživatelů požadavky na registraci, tím získává informaci o jejich aktuální poloze (IP

adresa, port a uživatelské jméno) tyto informace ukládá do lokalizační databáze (location database). V lokalizační databázi v tomto případě došlo k namapování adresy *sip:bob@b.com* k adrese *sip:bob@1.2.3.4:5060*. Lokalizační databáze je pak užívána proxy serverem a pokud proxy server obdrží žádost pro *sip:bob@b.com*, vyhledá v lokalizační databázi záznam *sip:bob@1.2.3.4:5060* a tam pošle žádost. Registrar server je velmi často implementován pouze jako logická část SIP serveru, jelikož je těsně svázán s proxy serverem.[6]

1.3 Registrace a přesměrování

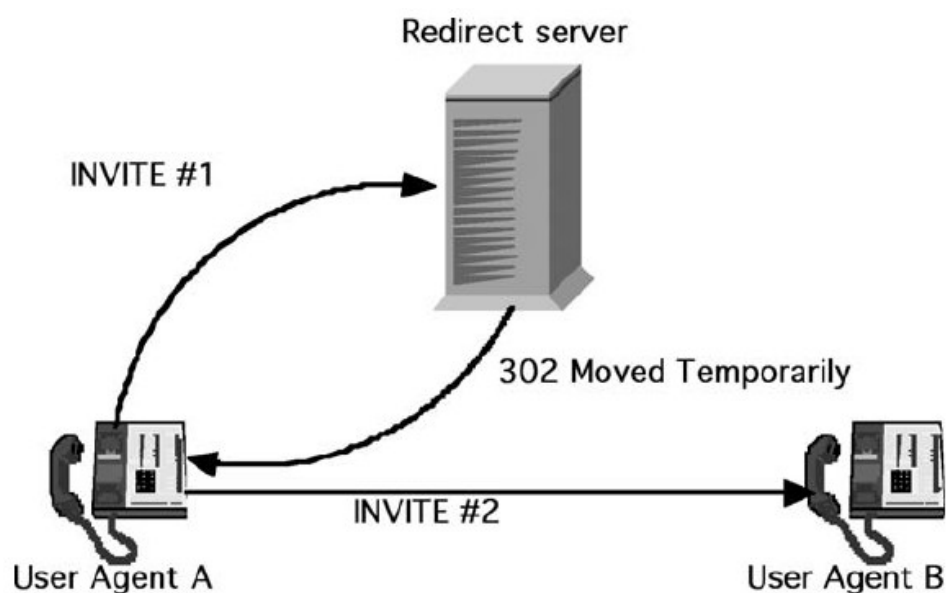
Další obrázek znázorňuje typickou SIP registraci. Zpráva REGISTER je vyslána do registrar serveru a obsahuje adresu záznamu *sip:jan@iptel.org* a kontaktní adresu *sip:jan@1.2.3.4:5060*, kde 1.2.3.4 je IP adresa telefonu. Registrar zaznamenává tyto informace do lokalizační databáze. Pokud registrace proběhla správně, tak registrar server posílá odpověď 200 OK a proces registrace je ukončen.



Obr. 2: Registrar server [6]

Registrace má omezenou dobu platnosti, ta je určena v hlavičce kontaktu, pokud ji neobnoví bude nedostupný.

Redirect server umožňuje přijímat požadavky a odesílat zpět odpovědi, které obsahují lokaci uživatele. Redirect server vyhledá dané požadavky v lokalizační databázi vytvořené díky němu. Poté vytvoří seznam aktuálních lokací uživatele a přešle je odesílateli v odpovědi označené díky třídě 3xx. Odesílatel v tuto chvíli dostane seznam destinací a odesílá další požadavky přímo na ně. [6]



Obr. 3: Redirect server a přesměrování hovorů [6]

1.4 Metody a odpovědi

SIP komunikace je tvořena zprávami, které jsou obvykle přenášeny v samostatných UDP datagramech. Každá zpráva obsahuje hlavičku zprávy (header) a vlastní obsah (body). V prvním řádku zprávy je identifikován její typ. Známe dva typy zpráv :

- žádost (metoda),
- odpověď.

1.4.1 Žádost

Je určena především k inicializaci nějaké procedury, jako sestavení nebo ukončení spojení. Jde přesně o oznámení příjemci na nějaký požadavek. Odpovědi jsou použity pro potvrzení o přijetí žádosti o jejím zpracování a výsledek zpracování.

Typická SIP žádost vypadá následovně:

```
INVITE sip:7170@iptel.org SIP/2.0
Via: SIP/2.0/UDP 195.37.77.100:5060
Max-Forwards: 10
;From:      „jiri“      <sip:jiri@iptel.org>tag=76ff7a07-c091-4192-
84a0d56e91fe104f
To: <sip:jiri@bat.iptel.org>
Call-ID: d10815e0-bf17-4afa-8412-d9130a793d96@213.20.128.35
CSeq: 2 INVITE
Contact: <sip:213.20.128.35:9315>
User-Agent: Windows RTC/1.0
Proxy-Authorization: Digest username=„jiri“, realm=„iptel.org“,
algorithm=„MD5“, uri=„sip:jiri@bat.iptel.org“,
nonce=„3cef75390000001771328f5ae1b8b7f0d742da1feb5753c“,
```

```
response=„53fe98db10e1074b03b3e06438bda70f“
Content-Type: application/sdp
Content-Length: 451
v=0
o=jku2 0 0 IN IP4 213.20.128.35
s=session
c=IN IP4 213.20.128.35
b=CT:1000
t=0 0
m=audio 54742 RTP/AVP 97 111 112 6 0 8 4 5 3 101
a=rtpmap:97 red/8000
a=rtpmap:111 SIREN/16000
a=fmtp:111 bitrate=16000
a=rtpmap:112 G7221/16000
a=fmtp:112 bitrate=24000
a=rtpmap:6 DVI4/16000
a=rtpmap:0 PCMU/8000
a=rtpmap:4 G723/8000
a=rtpmap: 3 GSM/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
```

První řádek nám určuje o jakou zprávu se jedná, v tomto případě jde o zprávu INVITE, jež je používána k sestavení spojení. URI na první řádce — *sip:7170@iptel.org* se nazývá Request URI a obsahuje URI dalšího skoku zprávy (next hop). V tomto případě bude hostitelem *iptel.org*. *Request URI* je tedy aktuální adresát požadavku.

V hlavičce je obsaženo jedno nebo více polí *Via*, které jsou použity pro záznam cesty žádosti. Ty jsou poté použity k směřování SIP odpovědi tou cestou, kterou byly odeslány. INVITE zpráva, která je v příkladu výše obsahuje jedno pole *Via*, které bylo vytvořeno UA. Lze z něj určit, že UA používá IP adresu *195.37.77.100* a port *5060*.

From a To v hlavičce identifikuje odesílatele (volajícího) a příjemce (volaného). Pole From obsahuje parametr tag, který bude popsán dále. Call-ID, jeho cílem je identifikovat zprávy náležející jednomu volání. Tyto zprávy mají stejný identifikátor Call-ID. Cseq je využito k správě pořadí požadavků, protože žádosti mohou být odeslány nespolehlivým přenosem, který může způsobit zpřeházení zpráv. V hlavičce je pole Contact obsahující IP adresu a port, na kterém odesílatel očekává další žádosti odesílané volaným.

Hlavička zprávy je oddělena od těla zprávy prázdným řádkem. Tělo zprávy žádosti INVITE obsahuje popis typu média vyhovující odesílateli a kódované v SDP.[6]

1.4.2 Specifikace žádostí (metod)

Žádosti resp. metody jsou specifikované v RFC 3261 následovně:

INVITE – žádost o inicializaci spojení nebo změnu parametrů probíhajícího spojení,

ACK - tato zpráva potvrzuje přijetí odpovědi na žádost INVITE.

BYE - zpráva je užívána k ukončení spojení některou ze stran,

CANCEL - je použit ke zrušení sestavovaného spojení, když volaný ještě nepotvrdil žádost INVITE a volající chce zrušit sestavování spojení,

REGISTER - smyslem žádosti je sdělit aktuální polohu uživatele. V této zprávě je přenášena informace o aktuální IP adrese a portu, na kterém může být uživatel zastížen. Registrace jsou časově limitovány a potřebují periodicky obnovovat,

OPTIONS - je žádost o zaslání schopností (vlastností) serveru nebo UA. [6]

1.5 Odpovědi

Pokud UA nebo Proxy server obdrží žádost, odešlou odpověď, každá žádost musí být zodpovězena, kromě žádosti ACK. Odpověď může vypadat následovně:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.1.30:5060;received=66.87.48.68
From: sip:sip2@iptel.org
To: sip:sip2@iptel.org;tag=794fe65c16edfdf45da4fc39a5d2867c.b713
Call-ID: 2443936363@192.168.1.30
CSeq: 63629 REGISTER
Contact: <sip:sip2@66.87.48.68:5060;transport=udp>;q=0.00;expires=120
Server: Sip EXpress router (0.8.11pre21xrc (i386/linux))
Content-Length: 0
Warning: 392 195.37.77.101:5060 „Noisy feedback tells:
pid=5110 req_src_ip=66.87.48.68 req_src_port=5060 in_uri=sip:iptel.org
out_uri=sip:iptel.org via_cnt==1“
```

Je zřejmé, že odpovědi jsou velmi podobné žádostem. První řádek však obsahuje verzi protokolu (SIP/2.0), kód odpovědi (reply code). Kód odpovědi je celé číslo z rozsahu 100 až 699 a označuje typ odpovědi, kterých je celkem 6 tříd a daly by se charakterizovat tímto způsobem:

- **1xx** tzv. informativní zprávy, které jsou použity pro odpovědi, které jsou stále zpracovávány a na jejich základě musí odesílatel zastavit posílání dané žádosti. Obvyklou odpovědí je 100 (Trying) a od UA se odesílají odpovědi s kódem 180 (Ringing), což oznamuje zvonění,
- **2xx** označovány za pozitivní odpovědi, které vyjadřují poslední odpovědi na zpracování dané žádosti. Odpovědi 200 – 299 znamenají úspěšné zpracování požadavků. Např. odpověď 200 OK je poslána pokud uživatel akceptoval žádost INVITE.

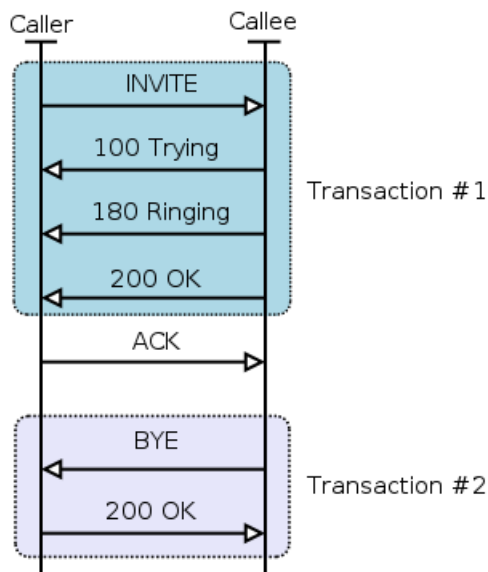
- **3xx** tyto kódy jsou určeny k přesměrování, jde o zaslání informace o nové poloze uživatele. Pokud z nějakého důvodu selže zpracování žádosti, vyšle proxy server odpověď volajícímu na požadavek o přesměrování s vloženou informací, které místo má volající kontaktovat, může jít jak o proxy server tak i o aktuální poloze volajícího. Tyto žádosti jsou konečné.
- **4xx** tyto odpovědi znamenají nějaký problém na straně odesílatele, jsou to konečné odpovědi a vyjadřují chybu v syntaxi,
- **5xx** zde je problém na straně serveru, žádost je možná v pořádku, ale server selhal při jejím zpracování, a proto by měl klient přeposlat žádost znovu,
- **6xx** tato žádost je vyslána, když žádost nemůže být vyřešena na žádném serveru, tato odpověď je posílána převážně serverem když má informaci o konkrétním uživateli, např. UA vysílá 603 Decline response, který odmítá žádost o sestavení spojení,

Tabulka 2: Třídy odpovědí [6]

kód	příklad
1XX	100 Trying 180 Ringing 183 Session Progress
2XX	200 OK 202 Accepted
3XX	300 Moved 302 Multiple Choices 305 Use Proxy
4XX	401 Unauthorized 403 Forbidden 415 Unsupported Media Type 486 Busy Here 428 Use Identity Header
5XX	501 Not Implemented 503 Service Unavailable
6XX	600 Busy Everywhere 603 Decline

SIP transakce – i když jsou SIP zprávy posílány sítí nezávisle, jsou obvykle uspořádány do transakcí agenty UA a určitými typy proxy serverů. Transakce je sekvence SIP zpráv, které se vyměňují mezi síťovými SIP prvky, obsahují jednu žádost a odpovědi, které se k ní vztahují, což může znamenat jednu nebo i více prozatímních odpovědí a i konečných odpovědí. Pokud byla transakce zahájena INVITE žádostí, pak obsahuje i zprávu ACK, ale jen pokud nebyla finální odpověď zpráva 200 OK. Za toto jsou odpovědní převážně UA, dokud nepřijmou ACK.

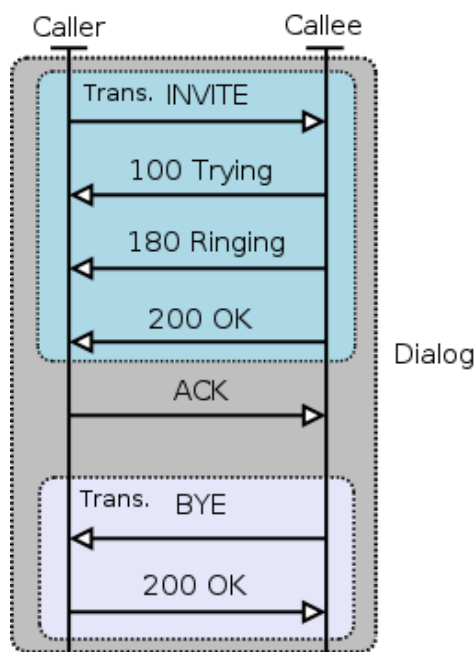
SIP entity, které sledují tyto transakce se nazývají stateful, čili se záznamem o stavech. Vytvořený stav je spojován s transakcí a je v paměti po celou dobu transakce. Aby mohl provést operaci, musí ze zprávy přečíst jednoznačný identifikátor transakce a porovnat jej s identifikátory právě probíhajících transakcí. Pokud existuje, doplní jej o další informace.



Obr. 4: Transakce [21]

Obrázek 4. znázorňuje zprávy zařazené do transakcí během konverzace dvou UA.

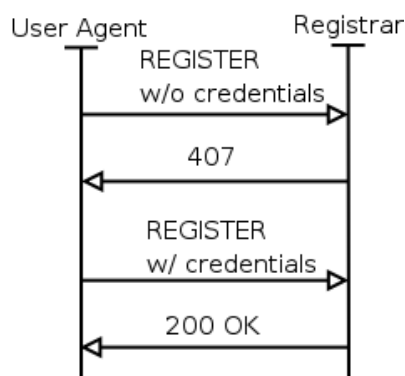
Dialog – je to soubor SIP zpráv peer-to-peer mezi dvěma UA, které mají vzájemnou spojitost a popisují řazení a směrování zpráv navzájem. Dialogy jsou identifikované pomocí pole Call-ID, From a To. Zprávy, které mají tyto tři identifikátory stejné, náleží jednomu dialogu. Více méně je dialog posloupnost transakcí. Obr.5 ukazuje, které zprávy patří jednomu dialogu. [6]



Obr. 5: Dialog [21]

1.6 Registrace na SIP serveru

Na registrar serveru se uživatelé musí registrovat sami, kvůli jejich dosažitelnosti. Ta se skládá ze zprávy REGISTER, po které následuje odpověď 200 OK, kterou odešle registrar server po úspěšném zalogování. Může se také stát, že UA dostane negativní odpověď 401 nebo 407 znamenající, že tato registrace není oprávněná. Toto je znázorněno na obr.6.



Obr. 6: Registrace [21]

Autentizace je základ bezpečnosti, která v SIPu díky protokolu HTTP (jeho ideový rodič), používá HTTP Digest. Autentizace rozdělujeme na:

- mezi uživateli (user-to-user) – registrační server je koncový příjemce požadavku, a proto je použita metoda User-to-user, pokud zpráva neobsahuje potřebné údaje, je zaslána zpět odpověď 401 Unauthorized a hlavička WWW Authenticate obsahuje výzvu.
- mezi proxy serverem a uživatelem (proxy-to-user). Pokud potřebuje proxy server ověřit uživatele učiní tak díky žádosti spojenou s ověřením klienta a provede se tak díky 407 Proxy Authenticate Required. Klienti doplní do požadavku hlavičku Proxy-Authorization údaji a celá registrace je zakončena odpovědí 200 OK. [6]

2 PROGRAMY PRO KOMUNIKACI

Zde bych chtěl napsat něco teorie o programech, které budou použity.

2.1 OpenSER

2.1.1 Historie projektu

Na podzim roku 2002 byl vytvořen společností FhG Fokus program SER (SIP Express Router). SER je v této době asi nejlepší implementace SIP Proxy. Byl vytvořen jen pár měsíců po vytvoření RFC3261 (SIP – vydán v létě roku 2002), který je dodnes hojně využíván v IP telefonii, Podpora pro SIP byla do SERu přidána v květnu 2003 a SER se stal nejlepší SIP Proxy.

SER začal být velmi populární a ukázalo se, že je velmi výkonný a komerčně využitelný. I možná kvůli této vlastnosti se projekt rozdělil a v roce 2004 se část týmu přesunula i se značkou SER do nově založené společnosti iptel.org, která moc nespolečovala s komunitou. Dva hlavní vývojáři



SER z původních pěti založili v roce 2005 nekomerční opensource projekt OpenSER. Tyto rozdělené projekty nadále pracovaly odděleně, každá na svém projektu. Pak se kvůli

sporům o obchodní značku SER přejmenoval z OpenSER 28.7.2008 na Kamailio. V tuto dobu se rozhodl hlavní vývojář OpenSERu Bogdan-Andrei Iancu, majitel domény openser.org, vytvořit další větev s názvem OpenSIPS. A tak 4. srpna 2008 vznikl projekt s názvem OpenSIPS, bylo to ani ne týden po vzniku projektu Kamailio. Během chvilky z webu prakticky zmizel veškerý obsah openser.org, včetně dokumentace k OpenSERu. Stále je nastaveno přesměrování ze stránek openser.org na opensips.org. Projekt Kamailio byl neznámý, vůbec se nevědělo co se děje, protože vyhledávače o něm vůbec nevěděly.



Nakonec se obě strany usmířili a dne 4. listopadu 2008 byl oznámen vznik projektu SIP Router, který by měl sjednotit Kamailio a SER. Nejdříve se upraví oba projekty, aby bylo možné použít stejný konfigurační soubor pro obě varianty, aby mohly sdílet společný kód. První krok je sjednotit kód jádra a transakční modul (tm). Jako základ se mají použít moduly SERu a do nich se budou portovat vybrané vlastnosti OpenSERu (pseudo-proměnné, konstrukce case, while apod.). Části kódu ze SERu se pak budou portovat do modulu z OpenSERu.[42]

2.1.2 Co je to OpenSER

Je to flexibilní serverová aplikace založená na SIP protokolu (RFC 3261). Důležitou vlastností je vytvořit přepojení pro VoIP telefonní systém. Je to svobodný software licencovaný GNU GPL. Hlavní vlastností OpenSERu je vytvoření robustního a vyváženého SIP serveru. Je používám převážně jako SIP-proxy a, registrar server, jako lokační nebo aplikační server nebo dokonce i jako brána pro služby, jakými jsou instant-messaging (SMS, Jabber).

OpenSER může být použit jak v malých systémech, např. uvnitř systému některých DSL routerů, tak i ve velkých společnostech s miliony uživatelů. Umožňuje i na slabších počítačích zvládat velké zátěže hovorů, např. na dvouprocesorovém počítači i několik tisíc hovorů za vteřinu. Je to program napsaný programovacím jazykem C a pracuje na systémech založených na UNIXu a Linuxu. Je řízen silným skriptovacím jazykem a je velmi modulární, díky přídatným modulům, o které je možné jej rozšířit. Podporuje IPv4 i IPv6, UDP a TCP (TLS) transporty. Umožňuje dokonce podporu pro více domén a ENUM. Výborná je podpora NAT průchodů díky nathelper, rtpproxy nebo mediaproxy modulů. Umožňuje spojení s databází MySQL.

Je obsažen snad v každé Linuxové distribuci. V těch sice nebude nejnovější verze, tu lze stáhnout na oficiálních stránkách projektu, jak <http://www.kamailio.org> tak i <http://www.opensips.org>.

2.2 Asterisk

Oficiálně jde o hybrid řešení TDM a packet voice PBX, jde o IVR platformu (Interactive Voice Response – automatický „odpovídač“ ovládan převážně tónovou volbou DTMF nebo pomocí hlasu) s funkcí ACD (Automatic Call Distribution – automatické rozdělování hovorů na čísla volajícího, časové podmínky). Jde o jednu z nejsilnějších flexibilních a rozšiřitelných řešení v oblasti telekomunikačního softwaru. Asterisk je open source pobočková ústředna (PBX – Private Branch eXchange), která pracuje na platformách Unix, Linux a Mac OSX. Je plnohodnotnou PBX, která je šířena také pod svobodnou licenci GNU GPL jako program OpenSER. Nesmírnou výhodou tohoto softwaru je, že za malé peníze (nákup vybavení) jej lze provozovat bez jakýchkoli vedlejších výdajů. Software je zdarma. Menší výdaje jsou také dosaženy díky jednodušší sestavení, konfiguraci a úpravám.



2.2.1 Využití Asterisku

Lze jej využít v různých dalších aplikacích díky podpoře:

- 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

2.2.2 Podpora kodeků

Asterisk podporuje následující kodeky:

- ITU G.711 a-law – 64 kbps, použití Evropa
- ITU G.711 u-law – 64 kbps, použití US
- ITU G.723.1 – 5.3/6.3 kbps, frame size 30 ms
- ITU G.726 – 16/24/32/40 kbps
- ITU G.729 – 8 kbps, frame size 10 ms
- GSM - 16 kbps Full Rate, frame size 20 ms
- ADPCM
- iLBC – internet Low Bitrate Codec. Volně dostupný kodek vhodný pro hlasovou komunikaci přes IP. Je vytvořen pro úzkopásmový hovor přenášený v užitečné zátěži přenosovou rychlostí 13.33 kbps s frame size 30 ms (399 bitů paketizováno

v 50 bytech) a přenosovou rychlostí 15.2 kbps s frame size 20 ms (303 bitů patetizováno v 38 bytech). Kodek umožňuje degradaci hlasové kvality v případě ztráty přenosových rámců, která nastane v případě spojení se ztrátou nebo zpožděním IP paketů.

- LPC10 – Linear Predictive Coding. Stejně jako kodek LPC ale s počtem 10 výpočtů předvídaných koeficientů. Používáno pro úzkopásmová spojení. Používá techniky zdrojového kódování. Tedy místo vysílání hovorového signálu je vypočítáván nejvhodnější signál mezi originálním signálem a filtrem. Poté jsou nejlépe odpovídající parametry tohoto filtru vyslány na dekodér. LPC dekodér použije tyto parametry ke generování syntetického hovoru, který je více či méně podobný originálnímu signálu. Výsledek je srozumitelný, ale jakoby „strojově“. Výstupní přenosová rychlost je 2.4 kbps.
 - Speex – open-source hlasový kodek (<http://www.speex.org>). Cílem je snížení bariér aplikací záznamu hovoru poskytnutím alternativ k drahým patentovaným hlasovým kodekům. Kodek je dobře adaptovatelný na Internetové aplikace a poskytuje užitečné vlastnosti, které nejsou dostupné ve většině ostatních kodeků.
- [7]

2.3 Porovnání OpenSERu a Asterisku

V oblasti VoIP technologie jsou to dva velmi silné programy, které umožňují špičkové služby s malými náklady. Následuje menší porovnání v některých oblastech.

2.3.1 Architektura

Asterisk je tzv. „back to back“ uživatelský agent (B2BUA), zatímco OpenSER je SIP (Session Initiation Protocol) proxy. To je hlavní rozdíl v architektuře, SIP proxy architektura je rychlejší než B2BUA, protože se zabývá pouze signalizací. Na druhé straně B2BUA architektura dokáže být velmi pomalá díky tomu, že je využita jako SIP proxy pro překládání kodeků z jednoho na druhý (např. G.729 <-> G.711, překlad protokolu SIP na H.323 a opačně). I některé služby související s médii jako IVR, Queuing nebo překlad z textu do řeči dokáží stroj, na kterém B2BUA běží, velmi zatížit.

2.3.2 NAT Traversal

OpenSER lépe pracuje za NATem než Asterisk. Může posílat média od svého zákazníka přímo k poskytovateli pomocí modulů nathelper, rtpproxy nebo mediaproxy. Řízení manipulace SIP protokolů povolují speciální případy, jako např. pro uživatele za jedním NAT zařízením, kdy je možné posílat média packety přímo mezi nimi.

2.3.3 Load Balancing

OpenSER má specifické rozdělení výkonu díky transformačním algoritmům. Tuto možnost Asterisk nemá, pouze jako doplňkovou službu.

2.3.4 Nízkoúrovňové přístupy k SIP hlavičkám a transakcím

OpenSER přistupuje k protokolu na nižší úrovni, proto může zvládnout všechny požadavky a odpovědi. Čili je možné většinu času strávit překládáním mezi dvěma nekompatibilními verzemi SIP protokolu, manipulacemi se SIP hlavičkami, žádostmi a odpověďmi. Je to důležitá vlastnost, pokud používáte zařízení od různých výrobců, které mohou být vzájemně odlišné.

2.3.5 Integrace s Radius, Diametr a LDAP

OpenSER vytvořilo integraci s Radius, Diametr a LDAP. Toto je možné i na Asterisku, ale implementace pro OpenSER je vyvíjena v C jazyce, integrována jako modul a je přímou součástí OpenSER distribuce (žádný perl, žádný python, žádné moduly třetích stran).

2.3.6 Služby médií

OpenSER je SIP proxy a není schopen jakékoli služby související s médii. Takže není možné pomocí něho vytvořit systémy jako hlasová pošta, IVR, TTS (text to speech) a rozpoznávání hlasu. Nicméně je možné tyto služby začlenit na stejné platformě pomocí samotných Media Serverů, jakými jsou Asterisk, Yate a FreeSwitch. Je to díky návrhu standardu (RFC3261).

2.3.7 Konektivita do PSTN

OpenSER potřebuje SIP bránu k připojení do PSTN sítě. Tento problém lze vyřešit instalací telefonní karty do serveru nebo Asterisk může být použit jako PSTN brána pro OpenSER.

2.3.8 Závěr porovnání

Dalo by se říci, že Asterisk a OpenSER se vzájemně doplňují. OpenSER řeší těžké nároky VoIP providerů, je schopný zvládnout velké objemy volání na zatížení SIP serveru, řešit pokročilé NAT scénáře a zabývat se SIP signalizacemi jako žádný jiný program. Asterisk je v poli B2BUA velmi silným PBX řešením. Je jednodušší na konfiguraci a zvládne nízké až střední objemy. Asterisk může být použit jako jediné řešení, zatímco OpenSER potřebuje architektonické součásti k práci. Asterisk je stále více oblíbený na PBX trhu, poněvadž více vyhovuje menším a středním poskytovatelům. Obvykle je OpenSER nasazován, jen pokud nastanou specifické potřeby jako vyrovnání zatížení nebo pokud vzniknou velké objemy, jako je více než tisíc registrovaných uživatelů.[27]

3 ENUM

3.1 Co je ENUM?

Telephone Number Mapping (*mapování telefonních čísel*, ENUM - tElephone NUmber Mapping) je soubor protokolů, které mají za cíl sjednotit telefonní systém s internetem za použití adres E.164 s DDDS a DNS. ENUM také označuje „E164 NUmber Mapping“. ENUM je protokol založený na DNS, který poskytuje vhodné služby, které pomáhají uskutečnit hovor, ale rozšiřuje i možnosti spojení o další DNS služby.[17]

ENUM nabízí volání zdarma na běžná telefonní čísla v jakékoli síti pomocí internetu a dokáže mnohonásobně ušetřit finanční prostředky. Na toto telefonní číslo si lze zaregistrovat speciální internetovou doménu, která ukazuje, jak se na dané telefonní číslo dovolat přes Internet. Poté je možné na toto číslo volat zdarma díky ENUM doméně. [37]

K využití výhody systému ENUM je potřeba si zaregistrovat ENUM doménu a zveřejnit tak ke svému číslu cestu, jak se na něj dovolat přes internet. [9]

3.2 Rozdíl mezi VoIP a ENUM

3.2.1 Běžný hovor přes VoIP

Volající na svém telefonu vytočí číslo, zde pomocí VoIP operátora je hovor směřován na ústřednu (pobočkovou nebo až na operátora). Tato ústředna má za úkol zjistit, zda-li je hovor směřován do veřejné telefonní sítě a pokud ano, předá hovor do této sítě a hovor je dále veden až k samotnému telefonu volaného, což znamená, že volající platí poplatky svému operátorovi za propojení a následný hovor.

Přes veřejnou telefonní síť je hovor propojen i tehdy, má-li jak volající, tak volaný IP telefon a existuje teoretická možnost spojit hovor po IP síti po celé jeho délce trasy. [9]

3.2.2 Hovor díky zaregistrovanému číslu pomocí ENUM

Po zaregistrování telefonního čísla do ENUM a vytvoření příslušného záznamu o tom, jak se na dané číslo dovolat pomocí VoIP do DNS, se proces změní. Volající vytočí číslo a příslušná ústředna se dotáže DNS, zda-li k volanému číslu existuje ENUM záznam. Pokud odpoví kladně, není již nutné hovor směřovat do veřejné telefonní sítě a hovor tak může být spojen po internetu přímo a volající neplatí za hovor žádné poplatky. Pokud záznam ENUM neexistuje, použije se standardní postup na propojení, čili pomocí

veřejných sítí.

ENUM tedy umožňuje vzájemné propojení VoIP telefonů díky nalezenému záznamu v DNS a tento hovor tak nadále nepotřebuje být přepojen přes PSTN síť. Ve své podstatě je ENUM jako telefonní seznam čísel, na něž se dá dovolat přes VoIP.

Hlavní rozdíl v obou metodách je ten, že u první metody volající neví o tom, že číslo by mohlo být i z VoIP telefonu. Nikdo mu to nezprostředkuje. Proto hovor bude přepojen díky veřejné síti. ENUM toto však zjistit dokáže a oba účastníci se mohou propojit vzájemně a volat si tak bez poplatků a zcela zdarma.[23]

3.3 Jak funguje volání s ENUM

ENUM je implementace standardu RFC3761, který stanovuje jak ukládat k běžnému číslu informace o službách na internetu. Definiuje jak mají být čísla zapsána do záznamu v e164.arpa. Postup uložení čísla do DNS je takovýto:

- máme telefonní číslo ve formátu +420212345678
- číslo se zbaví přebytečných nečíselných znaků a případných mezer
- číslo se obrátí a mezi každé číslo se vloží tečka: 8.7.6.5.3.2.1.2.0.2.4
- přidá se zmiňovaná doména e164.arpa: 8.7.6.5.3.2.1.2.0.2.4.e164.arpa

K této vytvořené doméně uložíme do DNS do speciálního NAPTR záznamu SIP adresu ve formátu: sip:uzivatel@domena. Pokud je proveden dotaz na doménové jméno, příslušný server vrátí odpověď se SIP adresou a pak může dojít k propojení telefonátu.[23]

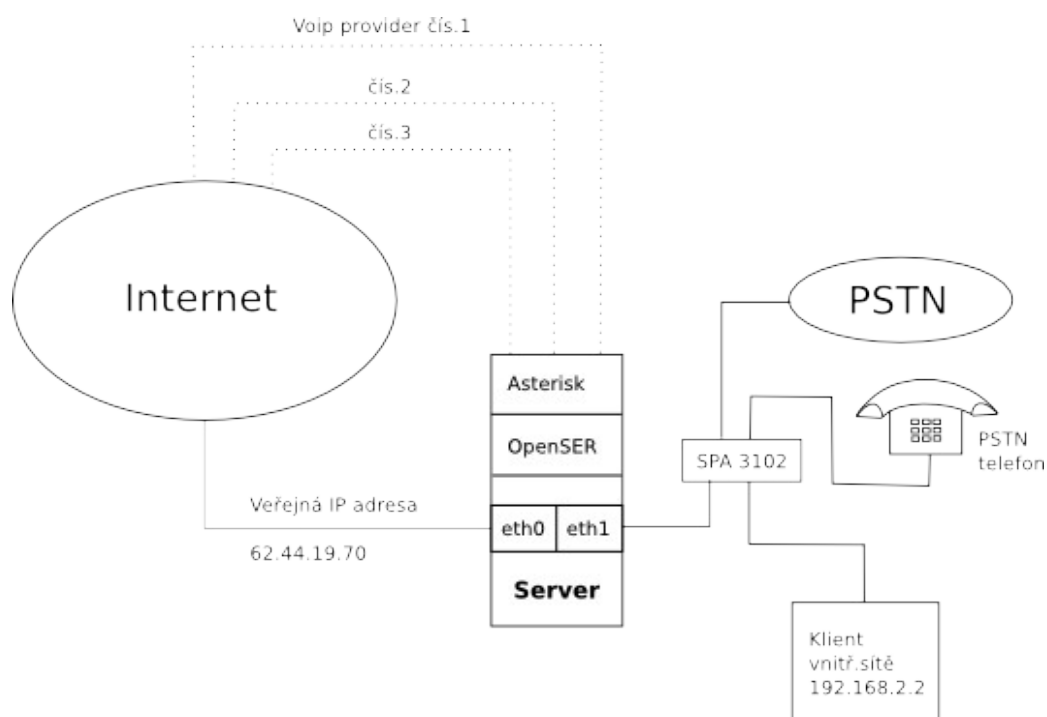
II. PRAKTICKÁ ČÁST

4 KONFIGURACE SIP SERVERU

Pro tuto diplomovou práci byla vybrána linuxová distribuce Debian 5.0 (označení Lenny), testing větev, 64bit. Tuto distribuci používám pro soukromé účely, proto jsem se rozhodl diplomovou práci provádět na tomto systému. Jádro bylo použito standardní, a to *2.6.26-1-amd64*.

Zde je popsána praktická část, kdy bude naznačen postup konfigurace VoIP serveru. Pro začátek bude popsána instalace a konfigurace OpenSER serveru, poté bude následovat instalace a konfigurace Asterisku.

Zapojení vypadá následovně (Obr.7). Fyzický stroj, označený jako „Server“, bude sloužit nejen jako server pro Asterisk a OpenSER, ale bude zároveň plnit i funkci IP routeru. V tomto stroji jsou nainstalovány 2 síťové karty, přičemž jedna z nich je přímo propojen do internetu (veřejná IP adresa) a přes druhou síťovou kartu je naroutováno připojení k SPA3102 VoIP bráně, která následně slouží jako další router pro dalšího klienta sítě (v mém případě můj osobní notebook). Toto řešení bylo navrženo hlavně z důvodu, aby příchozí pakety UDP z internetu se neNATovaly přes nějakou bránu a byl jim umožněn přímý přístup do serveru. Měl jsem navrženo i další řešení s připojením internetu (veřejné IP) do wifi-routeru, bohužel se nepodařilo přeposílat pakety UDP mezi wifi-routerem a serverem, tudíž se nedalo ani přihlásit na softwarovém telefonu.



Obr. 7: Topologie domácího zapojení

4.1 Instalace a konfigurace OpenSERu

4.1.1 Instalace

Instalace OpenSER se provede přidáním balíčku z mirroru debianu, který je uložen přímo v:

```
/etc/apt/sources.list
```

Zde je aktuálně uložen a nabízen balíček OpenSER v následující verzi:

```
tomik@debian:~$ aptitude show openser
Balík: openser
Stav: nainstalovaný
Automaticky instalovaný: ne
Verze: 1.3.2-3
Priorita: volitelný
Sekce: net
Správce: Debian VoIP Team <pkg-VoIP-maintainers@lists.alioth.debian.org>
Velikost po rozbalení: 3965k
```

Jelikož projekt OpenSER se přejmenoval od 4. srpna 2008 na projekt OpenSIPS je možné, že jde o konečnou verzi tohoto software v tomto mirroru. Projekt OpenSIPS ale nadále pokračuje ve „šlápějích“ OpenSER programu, klade si za důraz, že bude vycházet pravidelně každého půl roku, bude probíhat rozsáhlé testování před každým hlavním stabilním vydáním, příspěvky budou vyřízeny a implementovány, za předpokladu, že budou akceptovány, do méně než jednoho měsíce a mnohé další.

Pro tuto diplomovou práci bude OpenSER program plně vyhovovat.

4.1.2 Konfigurace

K nastavení OpenSER bude potřeba nastavit tyto soubory:

```
- openser.cfg
- openserctlrc
```

Následně je nutné editovat soubory a upravit je podle následujících ukázek:

```
- soubor openser.cfg1
- soubor openserctlrc
```

```
# $Id: openserctlrc 4133 2008-05-08 10:39:51Z miconda $
#
# The OpenSER configuration file for the control tools.
#
# Here you can set variables used in the openserctl and openserdbctl
setup
```

¹ Souboru `openser.cfg` je věnována samostatná kapitola(viz níže).

```
# scripts. Per default all variables here are commented out, the control
tools
# will use their internal default values.

## your SIP domain
SIP_DOMAIN=tomikova_domena

## database type: MYSQL, PGSQL, DB_BERKELEY, or DBTEXT, by default none
is loaded
# If you want to setup a database with openserdbctl, you must at least
specify
# this parameter.
DBENGINE=MYSQL

## database host
DBHOST=localhost

## database name
DBNAME=openser

# database path used by dbtext or db_berkeley
# DB_PATH="/usr/local/etc/openser/dbtext"

## database read/write user
DBRWUSER=openser

## password for database read/write user
# DBRWPW="openserrw"

## database read only user
# DBROUSER=openserro

## password for database read only user
# DBROPW=openserro

## database super user
DBROOTUSER="root"
```

Pro tento soubor byly zapotřebí pouze tyto úpravy:

SIP_DOMAIN: název domény, kterou bude používat OpenSER.

DBENGINE: je potřeba vybrat MYSQL pro jeho správnou funkci s MySQL serverem, který musí být také nainstalován na počítač.

Ostatní specifikace jako DBHOST, DBNAME, DBRWUSER, DBROOTUSER lze pochopit přímo z nápovědy v konfiguračním souboru, co mají na starosti.

- Dále byl upraven tento soubor /etc/default/openser

Zde je dobré nastavit, aby se OpenSER spouštěl při startu, kolik mu případně chceme přidělit paměti, apod.

```
# OpenSER startup options
#
# Set to yes to enable openser, once configured properly.
RUN_OPENSER=yes
# User to run as
```

```
USER=openser
# Group to run as
GROUP=openser
# Amount of memory to allocate for the running OpenSER server (in Mb)
MEMORY=64
# Enable the server to leave a core file when it crashes.
# Set this to 'yes' to enable OpenSER to leave a core file when it
crashes
# or 'no' to disable this feature. This option is case sensitive and only
# accepts 'yes' and 'no' and only in lowercase letters.
DUMP_CORE=no
```

Po těchto úpravách bylo potřeba nainstalovat a zkonfigurovat MySQL databázi.

4.1.3 Instalace, nastavení MySQL + přidání uživatele do OpenSERu

Instalace se provede tímto způsobem:

```
tomik@debian:~$ sudo aptitude install mysql-server-5.0
tomik@debian:~$ sudo aptitude install openser-mysql-module
```

Druhý balíček přidá podporu MySQL pro OpenSER. Po nainstalování MySQL musí být provedena změna hesla pro super uživatele (root):

```
tomik@debian:~$ mysql -u root -p mysql
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 112
Server version: 5.0.51a-24 (Debian)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> update user set password=password('Nami_pozadovane_heslo') where
user='root';
Query OK, 3 rows affected (0.11 sec)
Rows matched: 3  Changed: 3  Warnings: 0

mysql> exit
Bye
```

Po provedené změně hesla je potřeba restartovat server MySQL:

```
tomik@debian:~$ sudo /etc/init.d/mysql restart
```

Nyní je potřeba vytvořit openser databázi v MySQL, to provedeme následovně:

```
tomik@debian-doma:/etc/default$ sudo openserdctl create
MySQL password for root:
INFO: test server charset
INFO: creating database openser ...
INFO: Core OpenSER tables succesfully created.
Install presence related tables? (y/n): y
INFO: creating presence tables into openser ...
INFO: Presence tables succesfully created.
Install tables for imc cpl siptrace domainpolicy carrierroute? (y/n): y
INFO: creating extra tables into openser ...
```

```
INFO: Extra tables succesfully created.
```

Dále bylo nutné nastavit heslo pro openerse uživatele v MySQL:

```
tomik@debian:~$ mysql -u root -p mysql
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 71
Server version: 5.0.51a-24 (Debian)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> update user set password=password('Nami_pozadovane_heslo') where
user='openseer' ;
Query OK, 2 rows affected (0.00 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> exit
Bye
```

Byl proveden opětovný restart služby MySQL a v této fázi je možné přidat uživatele:

```
tomik@debian:~$ sudo openseerctl add tomik_notas tomik
tomik_notas@tomikova_domena
MySQL_password for user 'openseer@localhost':
new user 'tomik_notas' added
```

Přičemž celý příkaz má tyto části:

```
sudo openseerctl add UZIV_JMENO HESLO EMAIL+DOMENA
```

4.1.4 Úprava `openseer.cfg`

Tento soubor je velice obsáhlý a umožňuje nejrůznější nastavení OpenSERu. Během psaní této diplomové práce jsem použil standardní `openseer.cfg` soubor, který je zprostředkovaný již po instalaci OpenSERu. Zde byly provedeny menší úpravy a nezbytné přidání, případně odkomentování některých položek a výsledný konfigurační soubor se dal použít pro nastartování OpenSERu. Bude zde popsáno část tohoto konfiguračního souboru a vysvětleny jednotlivé položky, co znamenají. Samotný `openseer.cfg` soubor bude přiložen v příloze.

```
##### Global Parameters #####

debug=3
# OpenSER obsahuje ladící informace, které lze stupňovat. Resp.lze
stupňovat jeho „informovanost“, je to díky číslům od 0-9. Kdy 9 znamená
velkou informovanost.

log_stderr=no
#Pro zobrazení výstupu musíme nastavit na YES.

log_facility=LOG_LOCAL0
```

```
fork=yes
#Označuje, zda-li má být OpenSER spouštěn na pozadí. (Yes atribut).

children=4
#Určuje počet procesů spuštěných zároveň, pro produkční využití, kdy je
nápor na server větší je možné toto číslo zvýšit.

listen=udp:62.44.19.70:5060
#Adresa serveru, na které běží OpenSER v této části je i port pro
komunikaci, standardní je port 5060 pro všechny SIP protokoly.

##### Modules Section #####

#set module path
mpath="/usr/lib/openser/modules/"

/* uncomment next line for MySQL DB support */
loadmodule "mysql.so"
loadmodule "sl.so"
loadmodule "tm.so"
loadmodule "rr.so"
loadmodule "maxfwd.so"
loadmodule "usrloc.so"
loadmodule "registrar.so"
loadmodule "textops.so"
loadmodule "mi_fifo.so"
loadmodule "uri_db.so"
loadmodule "uri.so"
loadmodule "xlog.so"
loadmodule "acc.so"
loadmodule "auth.so"
loadmodule "auth_db.so"

# ----- setting module-specific parameters -----

# ----- mi_fifo params -----
modparam("mi_fifo", "fifo_name", "/tmp/openser_fifo")
## Jedná se o modul, který zajišťuje transportní vrstvu pro FIFO
implementací pro management rozhraní. fifo_name parametr určuje z jakého
vytvořeného souboru se mají poslouchat a číst externí příkazy.

modparam("rr", "enable_full_lr", 1)
## Zpětný work-around pro starší klienty, kteří neumí korektně zacházet
s hlavičkami SIP protokolu.

modparam("rr", "append_fromtag", 0)
## Je-li zapnuto (1) připojí se k žádosti trasy žádost od někoho (from-
tag), které jsou užitečné proto, aby bylo vysledováno zda pocházejí od
volajícího nebo od volaného.

# ----- rr params -----
modparam("registrar", "method_filtering", 1)
## Kontaktní filtrování založené na podporovaných metodách by mělo být
prováděno v průběhu lookup(u).

# modparam("registrar", "append_branches", 0)
## Když má jedno jméno více kontaktů, pokud je nastavena 1, žádost-URI
bude přepsána s největším-q hodnocením kontaktu, zbytek bude připojen
k sip_msg a může být později využívána. Pokud je nastaven na 0, bude
```

přepsána jen žádost-URI na nejvyšší hodnotu. Zbytek zůstane nezměněn.

```
/* uncomment the next line not to allow more than 10 contacts per AOR */
#modparam("registrar", "max_contacts", 10)
## Omezení počtu kontaktů pro AOR (Address od Record)

# ----- uri_db params -----
modparam("uri_db", "use_uri_table", 0)
## Zajišťuje použití uri tabulky, místo uživatelské tabulky
modparam("uri_db", "db_url", "")

# ----- acc params -----
modparam("acc", "radius_config", "")
modparam("acc", "early_media", 1)
modparam("acc", "report_ack", 1)
modparam("acc", "report_cancels", 1)
modparam("acc", "detect_direction", 0)
## Kontrola řízení detekce sekvenčních žádostí, pokud je aktivovaná
(nenulová hodnota), sekvenční žádosti jsou řízeny opačně (od volajícího
k volanému) FROM a TO budou zaměněny (řízení bude zachované jako původní
žádost) Ovlivňuje všechny hodnoty od TO k FROM hlavičkám (body, URI,
username, domain,..)

modparam("acc", "failed_transaction_flag", 3)
## Nastavení počtu selhaných transakcí
modparam("acc", "log_flag", 1)
## Žádost ve které se nastavuje seznam transakcí přes syslog
modparam("acc", "log_missed_flag", 2)
## Žádost ve které se nastavuje seznam ztracených volání přes syslog
modparam("acc", "db_flag", 1)
## Žádost ve které se nastavuje seznam transakcí - specifické pro
databázi
modparam("acc", "db_missed_flag", 2)
## Žádost ve které se nastavuje seznam ztracených volání - specifické pro
databázi

# ----- usrloc params -----
modparam("usrloc", "db_mode", 2)
## V usrloc modulu lze využít databázi k stálému ukládání kontaktů. Pokud
je využívána databáze, kontakty zůstanou v databázi i po restartu
počítače nebo jeho nečekaném pádu. Jeho hlavní nevýhoda je časová
náročnost, proto usrloc modul implementuje 4 základní režimy:
#0 - kompletní vypnutí databáze, bude použita jenom paměť, restart
znamená ztrátu kontaktů. Použití jen pokud chceme opravdu rychlý usrloc,
stále kontakty nejsou důležité, nebo jsou poskytovány jinými prostředky.
#1 - Write-Through; všechny změny se odrazí také v databázi. Je to velmi
pomalý režim, ale za to spolehlivý. Pokud rychlost není prioritou
a potřebujeme udržet kontakty v usrloc po restartu nebo pádu serveru.
#2 - Write-Back; Jedná se o kombinace dvou předešlých. Všechny změny se
provádějí v paměti a v určitém časovém intervalu jsou zapsány do
databáze. Tento režim je velmi náročný, pokud je server v neustálém
vytížení.
#3 - Read-Only režim. Neposkytuje databázi. Kontakty jsou přečteny
z databáze, ale změny nejsou zapsány. Kontakty nepřežijí restart.
Použijte tento režim pokud chcete výhody režimu 0 s podporou databázových
funkcí. Pozor: Je vhodné aktualizovat kontakty před nasazením serveru do
ostrého provozu.

modparam("usrloc", "db_url", "mysql://openser:heslo@localhost/openser")
## Odkaz na MySQL databázi
```



```
# ----- auth_db params -----
modparam("auth_db", "calculate_ha1", yes)
## Tento parametr znamená, že má server očekávat text hesla v databázi
nebo v HA1 řetězci. Pokud je parametr nastaven na hodnotu 1, bude se
server domnívat, že sloupec ukazující na password_column obsahuje text
hesla a bude počítat s HA1 řetězcem za běhu.
#Pokud je parametr nastaven na hodnotu 0, pak server předpokládá, že
databáze obsahuje HA1 znaky přímo a nebude s nimi počítat. Pokud parametr
username obsahuje @doménu, pak sloupec ukazuje na password_column_2,
který bude místo něj použit.
#To zajišťuje, že ověření bude vždy fungovat při použití předvýpočtu HA1
řetězce, který není v závislosti na přítomnosti domény v username.

modparam("auth_db", "password_column", "password")
##Název sloupce, který obsahuje hesla

##### Routing Logic #####
route{

    if (!mf_process_maxfwd_header("10")) {
## Tato služba je využívána pro sledování kolikrát prošla SIP zpráva
SERem. Špatně nastavené openser.cfg může způsobovat zacyklení podmínek.
Základním pravidlem tohoto parametru je, že pokud vrací parametr pořád
hodnotu 'true' pak se musí zastavit toto zpracování vadných zpráv, aby to
nevedlo k zacyklení.
        sl_send_reply("483","Too Many Hops");
## Pokud OpenSER detekoval smyčku je potřeba nějakým způsobem říci
klientovi, že došlo k chybě. To zajišťuje parametr sl_send_reply, který
je obsažen v sl.so modulu. OpenSER se po odeslání zprávy o nic nestará,
na většině telefonů by měla být zobrazena textová zpráva o stavu.
        exit;
    }

    if (has_totag()) {
        # Sekvenční žádost v rámci dialogu by měla mít cestu k určení
record-routing
        if (loose_route()) {
            if (is_method("BYE")) {
                setflag(1); # nastavení počítání
                setflag(3); # ... právě když transakce selže
            }
            route(1);
        } else {
            if ( is_method("ACK") ) {
                if ( t_check_trans() ) {
                    # non loose-route, but stateful ACK; must
be an ACK after a 487 or e.g. 404 from upstream server
                    t_relay();
                    exit;
                } else {
                    # ACK without matching transaction ...
ignore and discard.\n");
                    exit;
                }
            }
            sl_send_reply("404","Not here");
        }
        exit;
    }
}
```

```
# Zpracování CANCEL žádosti
if (is_method("CANCEL"))
{
    if (t_check_trans())
        t_relay();
    exit;
}

t_check_trans();

# Ověřovat, pokud je od lokálního účastníka
##if (!(method=="REGISTER") && from_uri==myself)
##{
##    if (!proxy_authorize("", "subscriber")) {
##        proxy_challenge("", "0");
##        exit;
##    }
##    if (!check_from()) {
##        sl_send_reply("403","Forbidden auth ID");
##        exit;
##    }
##}
## consume_credentials();
## # caller authenticated
##}

# Záznam směřování
if (!is_method("REGISTER|MESSAGE"))
    record_route();

# seznam pouze INVITE (žádostí)
if (is_method("INVITE")) {
    setflag(1); # počítání
}
if (!uri==myself)
/* Odkomentujte, pokud je vyžadována podpora více-domén */
##if (!is_uri_host_local())
{
    append_hf("P-hint: outbound\r\n");
    # pokud máte nějakou vnitřní doménu s připojením TLS
    ##if($rd=="tls_domain1.net") {
    ##    t_relay("tls:domain1.net");
    ##    exit;
    ##} else if($rd=="tls_domain2.net") {
    ##    t_relay("tls:domain2.net");
    ##    exit;
    ##}
    route(1);
}

# Žádost na mou doménu

/* Odkomentovat, pokud chcete zapnout osobní server, a zakomentujte
následující if podmínku včetně route[2]*/
##if( is_method("PUBLISH|SUBSCRIBE"))
##    route(2);

if (is_method("PUBLISH"))
{
    sl_send_reply("503", "Service Unavailable");
}
```

```
        exit;
    }

    if (is_method("REGISTER"))
    {
        # Ověřování REGISTER žádostí (odkomentujte pro zapnutí
ověřování)
        ##if (!www_authorize("", "subscriber"))
        ##{
        ##    www_challenge("", "0");
        ##    exit;
        ##}
        ##
        ##if (!check_to())
        ##{
        ##    sl_send_reply("403","Forbidden auth ID");
        ##    exit;
        ##}

        if (!save("location"))
            sl_reply_error();

        exit;
    }

    if ($rU==NULL) {
        # žádost s neobsahujícím Username v RURI
        sl_send_reply("484","Address Incomplete");
        exit;
    }

    # Aplikovat základní DB aliasy (odkomentovat pro zapnutí)
    ##alias_db_lookup("dbaliases");

    if (!lookup("location")) {
        switch ($retcode) {
            case -1:
            case -3:
                t_newtran();
                t_reply("404", "Not Found");
                exit;
            case -2:
                sl_send_reply("405", "Method Not Allowed");
                exit;
        }
    }
    # Při směřování přes usrloc, také pro zmeškané hovory
    setflag(2);

    route(1);
}

route[1] {
    # Pro využití pomocných tras INVITE zpráv
    if (is_method("INVITE")) {
        t_on_branch("2");
        t_on_reply("2");
        t_on_failure("1");
    }

    if (!t_relay()) {
```

```
        sl_reply_error();
    };
    exit;
}
```

Nyní mohlo být provedeno nainstalování softwarových telefonů (programů) na zkoušku hovoru v síti. Předtím bylo potřeba nastartovat OpenSER tímto příkazem:

```
tomik@debian:~$ sudo /etc/init.d/openser start
Starting openser: openserListening on
                udp: 62.44.19.70 [62.44.19.70]:5060
Aliases:
```

4.1.5 Instalace, nastavení programů X-Lite, WengoPhone

Pro tyto účely byly využity 2 programy X-Lite a WengoPhone.

X-Lite:

Tento prográmeček není sice opensource, ale je alespoň freeware, na adrese <http://www.counterpath.net/X-Lite-Download.html> lze stáhnout X-Lite_install.tar.gz balík, který se rozbalí následujícím příkazem:

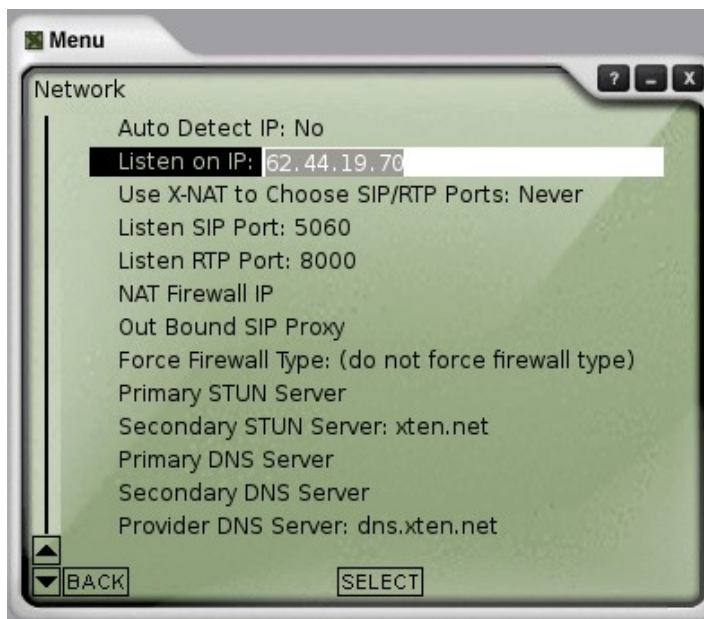
```
tar xfz X-Lite_install.tar.gz
```

Po rozbalení je potřeba spustit X-Lite program. Program vypadá následovně:



Obr. 8: X-Lite

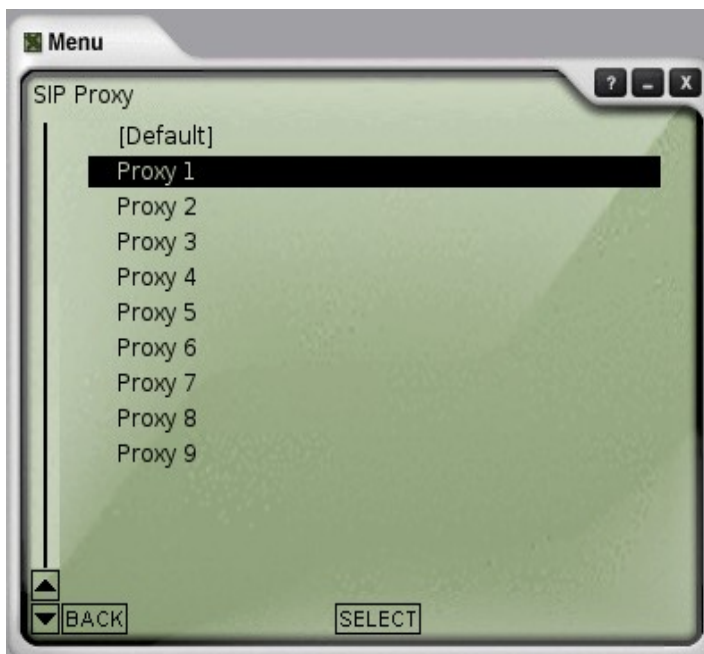
Poté je potřeba si nakonfigurovat X-Lite pro komunikaci, začíná se s nastavením sítě. Do tohoto menu se lze dostat z hlavního menu (Main menu), přes System Settings, zde je první položka zmiňované nastavení sítě (Network). Po jeho vybrání se zobrazí tyto položky:



Obr. 9: Nastavení sítě (X-Lite)

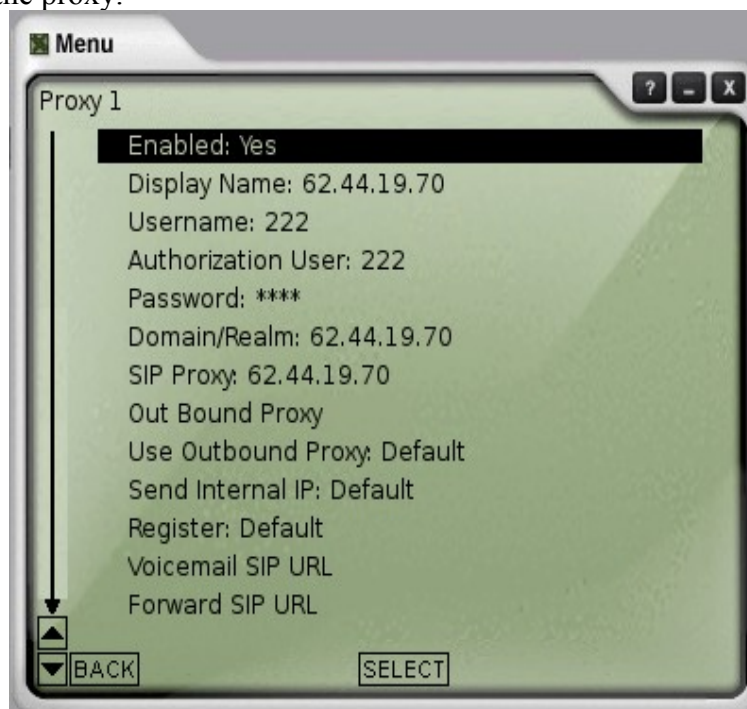
Zde je potřeba doplnit IP adresu počítače, na kterém je X-Lite spuštěný. Důležitý je i port pro SIP protokol, ale ten je standardně vyplněn.

Klikne se na tlačítko BACK a druhá položka je nastavení SIP proxy. Zde je podstatné vyplnit, ke kterému serveru se má X-Lite připojit. Standardně vypadá další obrazovka takto:



Obr. 10: Seznam proxy serverů (X-Lite)

Zde je vidět, že lze nakonfigurovat několik proxy, které mohou být použity. Pro tuto diplomovou práci bude v základu vybráno výchozí nastavení. Nyní bude provedeno nastavení samotné proxy:



Obr. 11: Nastavení proxy (X-Lite)

Enabled: Yes zajistí, že bude toto proxy nastavení aktivní.

Username: jméno uživatele.

Authorization User: zde už je přímo přihlašovací jméno uživatele, který byl přidán do databáze OpenSERu, čili jeho login.

Password: vyplněné heslo pro uživatele.

Domain/Realm: vyplněná doména, jak bylo nakonfigurováno v openserctrlrc souboru.

SIP Proxy, je IP adresa serveru, kde byl OpenSER nakonfigurován a spuštěn.

Tento program jsem se rozhodl nepoužívat, použil jsem jej jen ke zkoušce, zda bude proveden hovor z klienta na server. Měl jsem problémy s používáním zvukového zařízení na svém serveru (notebook), program hlásil, že nemůže najít žádné volně dostupné zařízení nebo že ho využívá nějaká služba. Bohužel i přesto, když jsem měl všechny možné programy, které využívají zvukového zařízení vypnuty, nepodařilo se zvuk rozjet. Na klientovi se mi X-Lite podařilo spustit a dokonce i fungoval. Fungovalo i přijetí hovoru a následné zavolání.

Další nepříjemností bylo celkem značné vytěžování procesoru počítače, a to jak klienta, tak i serveru. Což není moc dobré řešení pro každodenní používání.

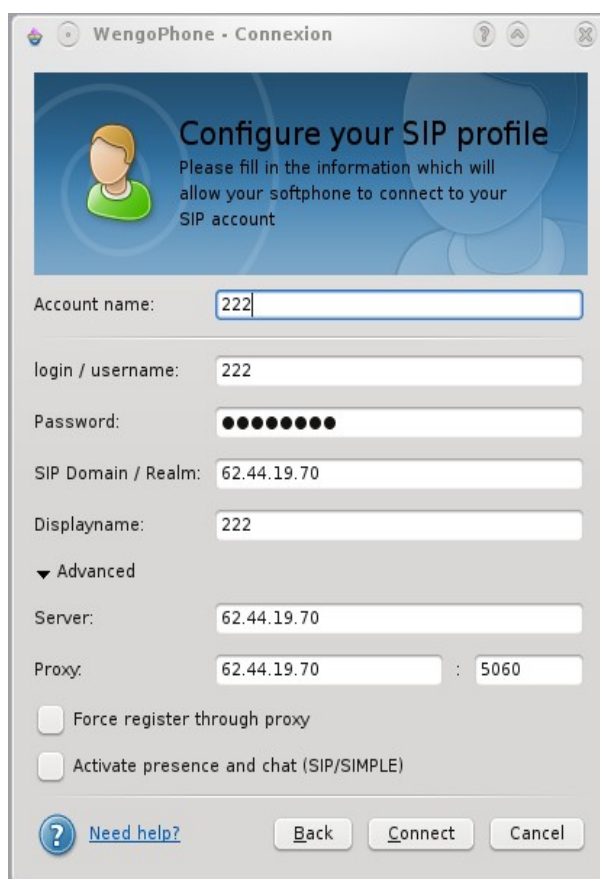
WengoPhone:

Jako lepší klientské řešení se zdá program WengoPhone od společnosti OpenWengo. Je to přívětivý prográmeček, který umožňuje plno nastavení, používání i přídavných protokolů, např. ICQ, Jabber, MSN, GoogleTalk, Yahoo apod. Čili je to multiprotokolový klient. Umožňuje konferenční hovory, posílání SMS zpráv (pro tuto službu je zapotřebí mít registrovaný účet u společnosti OpenWengo), všemožné úpravy kontakt listu, skupin atd.

Program je přímo v repozitáři Debianu, čili instalace byla velice snadná:

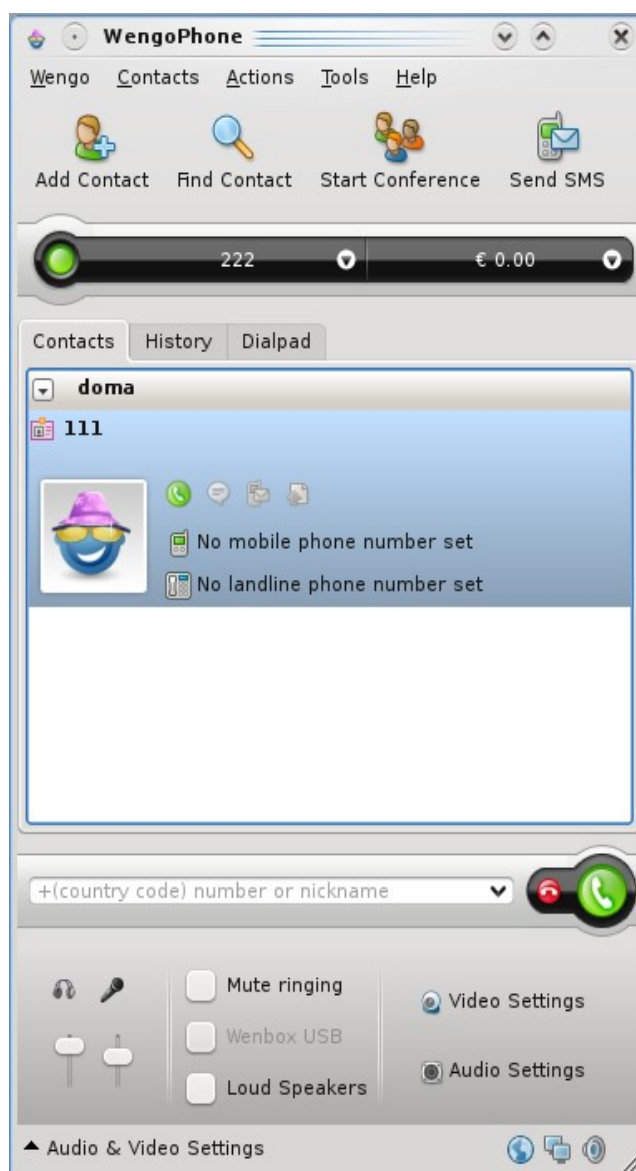
```
tomik@debian:~$ sudo aptitude install wengophone
```

První spuštění vyvolalo klikacího průvodce pro nastavení SIP profilu. Zde bylo nutné nastavit přihlašující údaje, doménu, server a proxy server. Viz následující obrázek.



Obr. 12: WengoPhone - nastavení účtu

Po přihlášení, resp. po kliknutí na tlačítko Connect se připojí program a hlavní okno vypadá následovně:



Obr. 13: Hlavní okno programu WengoPhone

5 REALTIME INTEGRACE PROGRAMŮ ASTERISK A OPENSER

Realtime integrací je myšleno, že oba programy budou využívat jedné společné databáze a tudíž i společných uživatelů, kteří se přidávají přímo do databáze OpenSER. V praxi by to mělo vypadat tak, že pokud se uživatel připojí na OpenSER na portu 5060, tak by měl Asterisk zvládnout zavolat v rámci realtime integrace na tohoto uživatele a opačně. Do souboru *openser.cfg* je potřeba přidat ***rewritehost*** položku, pro přesměrování na IP adresu Asterisku a jeho port, který bude nastaven na hodnotu 5065. Část pro přesměrování na Asterisk by měla vypadat následovně:

```
route[3] {
# -----
# ASTERISK Handler
# -----
    xlog("L_INFO", "Step 500 Entering ==== Route 3 (Asterisk) =====");
    rewritehostport("62.44.19.70:5065"); # INSERT YOUR Asterisk
GATEWAY IP ADDRESS and port here
route(1);
}
```

Program Asterisk bude využíván na propojení PSTN a VoIP operátora s mojí telefonní ústřednou a komunikaci mezi uživateli. Pro svou funkcionalitu je použit právě Asterisk, jelikož OpenSER nemá takové možnosti v používání při komunikaci a „vybavení“ ústředny jako Asterisk. Jde např. o aplikace uskutečňování konferenčních hovorů, různé použití VoIP bran (gateways) (SIP, H.323, IAX, MGCP), má vlastnost pobočkové ústředny (PBX), voicemail služby s adresářem, interaktivní hlasový průvodce (IVR) server a mnohé další.

5.1 Instalace, konfigurace Asterisku

Asterisk jsem opět použil přímo z repozitáře Debianu, kde je v následující verzi:

```
tomik@debian-doma:/ sudo aptitude show asterisk
[sudo] password for tomik:
Balík: asterisk
Stav: nenainstalovaný
Verze: 1:1.4.21.2~dfsg-3
Priorita: volitelný
Sekce: comm
Správce: Debian VoIP Team <pkg-VoIP-maintainers@lists.alioth.debian.org>
.....
```

Nainstalujeme jej standardním příkazem:

```
sudo aptitude install asterisk
```

Nyní je potřeba vytvořit databázi, která bude komunikovat přímo s OpenSER databází. To je kvůli tomu, pokud přidáme uživatele do OpenSER databáze, aby mohl být použit

i v programu Asterisk a jeho databázi. Čili budou vzájemně propojeny.

Zalogujeme se do MySQL databáze:

```
tomik@debian:~$ mysql -u root -p mysql
```

Zde změníme databázi:

```
mysql> USE opener
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

Je možné přidat položku pro heslo do hlasové schránky, která může být později přidána do Asterisku:

```
mysql> ALTER TABLE subscriber ADD vmail_password varchar(32);
Query OK, 4 rows affected (0,03 sec)
Records: 4 Duplicates: 0 Warnings: 0
```

Nyní můžeme vytvořit databázi Asterisku (zadávají se příkazy, které jsou uvedeny níže):

```
mysql> create database asterisk;
Query OK, 1 row affected (0,00 sec)
mysql> use asterisk;
Database changed
mysql> CREATE TABLE `voicemessages` (
->   `id` int(11) NOT NULL auto_increment,
->   `msgnum` int(11) NOT NULL default '0',
->   `dir` varchar(80) default '',
->   `context` varchar(80) default '',
->   `macrocontext` varchar(80) default '',
->   `callerid` varchar(40) default '',
->   `origtime` varchar(40) default '',
->   `duration` varchar(20) default '',
->   `mailboxuser` varchar(80) default '',
->   `mailboxcontext` varchar(80) default '',
->   `recording` longblob,
->   PRIMARY KEY (`id`),
->   KEY `dir` (`dir`)
-> ) ENGINE=InnoDB;
Query OK, 0 rows affected (0,03 sec)

mysql> CREATE VIEW vmusers AS
-> SELECT id as uniqueid,
->        username as customer_id,
->        'default' as context,
->        username as mailbox,
->        vmail_password as password,
->        CONCAT(first_name,' ',last_name) as fullname,
->        email_address as email,
->        NULL as pager,
->        datetime_created as stamp
-> FROM opener.subscriber;
Query OK, 0 rows affected (0,00 sec)

mysql> CREATE VIEW sipusers AS
-> SELECT username as name,
->        username,
```

```

-> 'friend' as type,
-> NULL as secret,
-> domain as host,
-> CONCAT(rpid, ' ', '<', username, '>') as callerid,
-> 'default' as context,
-> username as mailbox,
-> 'yes' as nat,
-> 'no' as qualify,
-> username as fromuser,
-> NULL as authuser,
-> domain as fromdomain,
-> NULL as insecure,
-> 'no' as canreinvite,
-> NULL as disallow,
-> NULL as allow,
-> NULL as restrictcid,
-> domain as defaultip,
-> domain as ipaddr,
-> '5060' as port,
-> NULL as regseconds
-> FROM opener.subscriber;

```

Query OK, 0 rows affected (0,00 sec)

Nyní musíme přidat práva pro MySQL uživatele pro plný přístup k databázi „asterisk“.

```

mysql> GRANT ALL ON asterisk.* to asterisk@localhost IDENTIFIED BY
'Heslo_ktere_pouzivame_pro_pripojeni_do_MySQL';

```

Query OK, 0 rows affected (0,00 sec)

5.1.1 Konfigurace hlavních souborů Asterisku

Zde budou uvedeny úpravy souborů Asterisku pro správné routování mezi oběma programy. Nejdříve upravíme soubor *sip.conf*.

```

[general]
context=users                ; Default context for incoming calls
allowoverlap=no              ; Disable overlap dialing support.
                              (Default is yes)

bindport=5065                 ; UDP Port na kterém běží Asterisk

bindaddr=0.0.0.0              ; IP adresa (0.0.0.0 prohledává všechny
                              IP adresy)

disallow=all
allow=ulaw
allow=alaw
allow=gsm

register => 111:heslo@62.44.19.70/111; registrace pevného čísla
Asterisku, pro propojení s OpenSERem

[users]

[222]
type=friend                   ;umožňuje jak příchozí tak i odchozí volání
username=222
userid=test2 <222>
host=dynamic                  ;bude použita IP z příchozího volání
context=users                 ;odkaz na kontext v extensions.conf

```

```
secret=heslo ;heslo

[111] ;definování uživatele 111 (Asterisk)
type=friend
user=111
host=62.44.19.70
realm=62.44.19.70
fromdomain=62.44.19.70
fromuser=111
secret=heslo
insecure=very
context=users ;odkaz na kontext v extensions.conf
authname=111
dtmfmode=RFC2833
canreinvite=no
username=111
disallow=all
allow=ulaw
```

Dále je potřeba přidat pár řádků do *extensions.conf*.

```
[general]
;
static=yes
writeprotect=yes ;Zabránění ladění dialplanu v konsoli Asterisku

; The "Globals" category contains global variables that
; can be referenced in the dialplan with ${VARIABLE} or
; ${ENV(VARIABLE)} for Environmental variable
; ${${VARIABLE}} or ${text${VARIABLE}} or any hybrid
;
[globals]
; =====
; ===== Dialplan =====
; =====
SERADDRESS=62.44.19.70:5060 ;IP adresa:port OpenSER programu
sounds_exten = /usr/share/asterisk/sounds/ ;cesty k zvuk. souborům
sounds_phonetic = /usr/share/asterisk/sounds/phonetic/
sounds_letters = /usr/share/asterisk/sounds/letters/
sounds_silence = /usr/share/asterisk/sounds/silence/

[users]
exten => _111,1,Dial(SIP/111,10) ;po zavolání na 111 použije číslo
; definované v sip.conf (111)
exten => _111,n,NoOP(Dojde to sem?) ;pouze výpis na konsoli Asterisku
exten => _111,n,goto(echo,s,1) ;provede skok na echo kontext
exten => _XXX,1,Dial(SIP/${EXTEN}) ;definované příchozí volání na
; trojmístná čísla v sip.conf

[echo]
exten => s,1,Playback(demo-echotest)
exten => s,n,Echo()
exten => s,n,Playback(demo-echodone)
exten => s,n,Goto(s,1)
```

5.2 Ukončení dosavadní myšlenky diplomové práce

Tyto úpravy mohly stačit k vzájemnému zkoušení v realtime integraci mezi oběma programy. Bohužel po několika denním zkoušení různých pravidel, jak v *openser.cfg* souboru, tak i různá nastavení v souborech Asterisku, nevedla tato cesta k zdárnému konci. Problém nastal už při samotném volání z jednoho čísla na druhé. V Asterisku jsem si nastavil pravidlo, že pokud někdo zavolá a 10 vteřin nikdo nebude zvedat telefon, tak se přehraje volajícímu hláška, že momentálně není nikdo dostupný. Vše fungovalo, pokud byl uživatel „natvrdo“ přihlášen do Asterisku přes port (v mém případě) 5065. Problém byl naopak u přihlášení uživatele na port 5060, čili OpenSER. Zde, nechápu proč, přesměrování nefungovalo žádným způsobem. Při řešení problému jsem zjistil, že je celkem problém tyto dva programy provozovat na jednom serveru současně. Vzájemná komunikace se někde ztrácela nebo bylo něco špatně nastaveno, ale svým úsilím jsem nezjistil, co by mohlo být v nepořádku. Proto jsem se rozhodl tuto cestu opustit a přenechat ovládání PBX čistě jen na programu Asterisk.

6 PBX ASTERISK

Jak již bylo napsáno v minulé kapitole, původní myšlenka propojení OpenSER a Asterisk programu nevedla ke zdárnému konci, a tudíž se budu nyní zabývat samotným řízením hovorů díky PBX Asterisk, která je již nainstalována na serveru, a proto není problém pokračovat. Bylo potřeba vypnout OpenSER program přímo v souboru */etc/default/openser* a změnit položku *RUN_OPENSER=yes* na *NO*. Tím se zajistí, že OpenSER se již nebude dále startovat. Dále zakomentovat řádek v *sip.conf* souboru v adresáři Asterisk *SERADDRESS=62.44.19.70:5060*, který odkazoval na OpenSER.

V rámci této diplomové práce mám řešit vzájemné volání jak z ústředny ven tak aby se jakékoli číslo dovolalo mně, a to jak čísla z PSTN sítě, tak i mobilní telefony. Bude zde následovat i nastavení a popis základních parametrů z webového rozhraní, kde je potřeba nastavit v rámci brány SPA3102, která umožňuje použití a přímé propojení s PSTN.

6.1 Základní ovládání Asterisku:

Pro spuštění konzole, kde se vypisují příkazy pro Asterisk nebo pro sledování výpisů (co se děje při volání), lze zapnout pomocí příkazu:

```
sudo asterisk -rvvv
```

Kde nám tři V označují „verbose“ mód, čili jak moc výpisů se bude zobrazovat v konzoli.

Další příkazy lze zadávat přímo v konzoli Asterisku, uvedu zde ty nejpoužívanější.

Tabulka 3: Základní příkazy v Asterisk konzoli

Příkaz:	Vysvětlení:
help	Nápověda (vypíše všechny příkazy)
show dialplan (NAZEV_DP)	Ukáže celý dialplan nebo výpis zadaného dialplanu
core set verbose 10	Nastavení verbose módu
dialplan reload	Pro nahrání změn v <i>extensions.conf</i> souboru
sip reload	Pro nahrání změn v <i>sip.conf</i> souboru
reload	To samé, jenom provede kompletní reload Asterisku
show uptime	Uptime Asterisku
restart when convenient	Restart Asterisku, když nebudou probíhat žádné hovory

6.2 SPA3102 brána

Toto zařízení od společnosti LINKSYS/Cisco umožňuje propojení s PSTN díky FXS portu a připojení analogového telefonu díky FXO portu, dále umožňuje přidat 1 VoIP operátora.

Umožňuje tyto funkce:

- router, gateway (analogový převodník)
- zapojení standardní tel. linky
- podpora protokolu SIP (v2)
- funkce potlačení echa
- podpora kodeků G.711, G.726, G.729, G.723.1
- konfigurace pomocí web prohlížeče
- automatickou funkci telefonní zálohy při výpadku internetu



Obr. 14: Linksys SPA3102

Díky rozsáhlému webovému rozhraní, lze nakonfigurovat SPA pro mnoho funkcí. Lze nastavit napevno jednoho VoIP operátora. Pokud bychom chtěli 2 operátory, je tato možnost díky zařízení SPA2102, bohužel toto zařízení neumožňuje připojení klasické pevné linky. To samozřejmě na SPA3102 lze řešit pomocí Asterisku a nakonfigurování více VoIP operátorů s voláním skrz tzv. prefixy.

Abychom mohli provádět úpravy v SPA3102, musíme nastavit IP adresu v rozsahu, v kterém je SPA, což je 192.168.0.0-255 rozsah. Samotné SPA je nastaveno na IP adresu 192.168.0.1. Pro mé účely bylo SPA v provedení jako router, kdy jsem na něj měl připojen i můj osobní notebook.

6.2.1 Nastavení SPA - Line1 tab

Pro toto nastavení jsem vyfotil jednotlivé položky ve webovém rozhraní a následně je okomentuji. Pro nastavení služeb je potřeba se přihlásit jako Admin. Vpravo nahoře postupně klikneme na **Admin login** → **Advanced** → **Line 1 tab**.

The screenshot shows the configuration page for 'Line 1' in a Cisco router. The page is divided into several sections, each with a title and a list of configuration parameters. The parameters are set to specific values, some of which are highlighted with red boxes in the original image. The sections and their settings are as follows:

- Line Enable:** yes
- Streaming Audio Server (SAS):**
 - SAS Enable: no
 - SAS DLG Refresh Intvl: 30
 - SAS Inbound RTP Sink: (empty)
- NAT Settings:**
 - NAT Mapping Enable: yes
 - NAT Keep Alive Enable: yes
 - NAT Keep Alive Msg: \$NOTIFY
 - NAT Keep Alive Dest: \$PROXY
- Network Settings:**
 - SIP ToS/DiffServ Value: 0x68
 - SIP CoS Value: 3 [0-7]
 - RTP ToS/DiffServ Value: 0xb8
 - RTP CoS Value: 6 [0-7]
 - Network Jitter Level: high
 - Jitter Buffer Adjustment: up and down
- SIP Settings:**
 - SIP Port: 5060
 - SIP 100REL Enable: no
 - EXT SIP Port: (empty)
 - Auth Resync-Reboot: no
 - SIP Proxy-Require: (empty)
 - SIP Remote-Party-ID: yes
 - SIP GUID: no
 - SIP Debug Option: full
 - RTP Log Intvl: 0
 - Restrict Source IP: no
 - Referor Bye Delay: 4
 - Refer Target Bye Delay: 0
 - Referee Bye Delay: 0
 - Refer-To Target Contact: no
 - Sticky 183: no
- Call Feature Settings:**
 - Blind Attn-Xfer Enable: no
 - MOH Server: (empty)
 - Xfer When Hangup Conf: yes
- Proxy and Registration:**
 - Proxy: 62.44.19.70
 - Outbound Proxy: (empty)
 - Use Outbound Proxy: no
 - Use OB Proxy In Dialog: no
 - Register: no
 - Make Call Without Reg: yes
 - Register Expires: 600
 - Ans Call Without Reg: yes
 - Use DNS SRV: no
 - DNS SRV Auto Prefix: no
 - Proxy Fallback Intvl: 3600
 - Proxy Redundancy Method: Normal
 - Voice Mail Server: (empty)
 - Mailbox Subscribe Expires: 2147483647

Obr. 15: Line 1 záložka - 1.část

Tyto hodnoty jsou pozměněny převážně pro VoIP hovory díky 802.cz a jejich návodu[41].

- **SIP Port: 5060**
- **Proxy: IP adresa serveru, kde běží Asterisk**
- **Use OB Proxy in Dialog: NO**
- **Register Expires: 600** (nutné pro 802.cz)
- **Ans Call Without Reg.: YES** (tato volba byla mnou nastavena, když jsem měl menší problémy s nastavením dialplanu. V podstatě Vám tato volba umožní volat,

bez Astersiku na PSTN po zmáčknutí # (i když # souvisí hlavně z nastaveným dialplanem viz. níže))

Subscriber Information			
Display Name:	<input type="text"/>	User ID:	<input type="text" value="spa"/>
Password:	<input type="password" value="*****"/>	Use Auth ID:	<input type="text" value="no"/>
Auth ID:	<input type="text"/>		
Mini Certificate:	<input type="text"/>		
SRTP Private Key:	<input type="text"/>		

Obr. 16: Line 1 záložka - 2.část

Zde je potřeba vyplnit User ID a heslo, které bude použito v souboru *sip.conf*. Tento soubor bude probrán v další kapitole.

Audio Configuration			
Preferred Codec:	<input type="text" value="G711a"/>	Silence Supp Enable:	<input type="text" value="no"/>
Use Pref Codec Only:	<input type="text" value="no"/>	Silence Threshold:	<input type="text" value="medium"/>
G729a Enable:	<input type="text" value="yes"/>	Echo Canc Enable:	<input type="text" value="yes"/>
G723 Enable:	<input type="text" value="yes"/>	Echo Canc Adapt Enable:	<input type="text" value="yes"/>
G726-16 Enable:	<input type="text" value="yes"/>	Echo Supp Enable:	<input type="text" value="yes"/>
G726-24 Enable:	<input type="text" value="yes"/>	FAX CED Detect Enable:	<input type="text" value="yes"/>
G726-32 Enable:	<input type="text" value="yes"/>	FAX CNG Detect Enable:	<input type="text" value="yes"/>
G726-40 Enable:	<input type="text" value="yes"/>	FAX Passthu Codec:	<input type="text" value="G711u"/>
DTMF Process INFO:	<input type="text" value="yes"/>	FAX Codec Symmetric:	<input type="text" value="yes"/>
DTMF Process AVT:	<input type="text" value="yes"/>	FAX Passthu Method:	<input type="text" value="NSE"/>
DTMF Tx Method:	<input type="text" value="INFO"/>	FAX Process NSE:	<input type="text" value="yes"/>
Hook Flash Tx Method:	<input type="text" value="None"/>	FAX Disable ECAN:	<input type="text" value="no"/>

Obr. 17: Line 1 záložka - 3.část

Zde je dobré vybrat preferovaný kodek (**Preferred Codec**) na volání **G711a** (i kvůli volání přes VoIP 802.cz).

- **DTMF Tx Method: INFO**

Dial Plan	
Dial Plan:	<input type="text" value="([2-9]xxxxxxxxS0 *xx **xx *xx* xx. *xxx <#,:>[2-9]xxxxxxxxS0<:@gw0> <#,:>"/>
Enable IP Dialing:	<input type="text" value="no"/>
Emergency Number:	<input type="text"/>

Obr. 18: Line 1 záložka - 4.část

Vložíme následující dialplan pro volání. Tento dialplan je převzat přímo ze stránek 802.cz a umožňuje volání na jakékoli čísla. Pro vytáčení pevné linky, bez použití SPA3102 je vybrán prefix #.

Dial Plan: `([2-9]xxxxxxxxS0|*xx|**xx|*xx*|xx.|*xxx|<#,:>[2-9]xxxxxxxxS0<:@gw0>|<#,:>xx.<:@gw0>)`

6.2.2 České oznamovací tóny

Jelikož jsou v SPA standardně nastaveny americké oznamovací tóny je dobré je přepsat, alespoň ty základní. Podklady jsem našel zde[38].

Info	System	SIP	Provisioning	Regional	Line 1	PSTN Line	User 1	PSTN User	Ser Login	basic	advance
Call Progress Tones											
Dial Tone:	425@-10;30(0.33/0.33/1,0.66/0.66/1)										
Second Dial Tone:	420@-19,520@-19;10(*0/1+2)										
Outside Dial Tone:	420@-16;10(*0/1)										
Prompt Tone:	520@-19,620@-19;10(*0/1+2)										
Busy Tone:	425@-10;10(0.33/0.33/1)										
Reorder Tone:	480@-19,620@-19;10(.25/.25/1+2)										
Off Hook Warning Tone:	480@-10,620@0;10(.125/.125/1+2)										
Ring Back Tone:	425@-10;60(1/4/1)										
Confirm Tone:	600@-16;1(.25/.25/1)										
SIT1 Tone:	985@-16,1428@-16,1777@-16;20(.380/0/1,.380/0/2,.380/0/3,0/4/0)										
SIT2 Tone:	914@-16,1371@-16,1777@-16;20(.274/0/1,.274/0/2,.380/0/3,0/4/0)										
SIT3 Tone:	914@-16,1371@-16,1777@-16;20(.380/0/1,.380/0/2,.380/0/3,0/4/0)										
SIT4 Tone:	985@-16,1371@-16,1777@-16;20(.380/0/1,.274/0/2,.380/0/3,0/4/0)										
MWI Dial Tone:	350@-19,440@-19;2(.1/.1/1+2);10(*0/1+2)										
Cfwd Dial Tone:	350@-19,440@-19;2(.2/.2/1+2);10(*0/1+2)										
Holding Tone:	600@-19;*(.1/.1/1,.1/.1/1,.1/9.5/1)										
Conference Tone:	350@-19;20(.1/.1/1,.1/9.7/1)										
Secure Call Indication Tone:	397@-19,507@-19;15(0/2/0,.2/.1/1,.1/2.1/2)										
VoIP PIN Tone:	600@-10;*(0/1/1,.1/.1/1,.1/.1/1,.1/.5/1)										
PSTN PIN Tone:	600@-10;*(0/.7/1,.2/.1/1,.2/.1/1,.2/.5/1)										
Feature Invocation Tone:	350@-16;*(.1/.1/1)										

Obr. 19: Regional – české oznamovací tóny

- Dial Tone : **425@-10;30(0.33/0.33/1,0.66/0.66/1)**
- Busy Tone : **425@-10;10(0.33/0.33/1)**
- Ring Back Tone : **425@-10;60(1/4/1)**

6.2.3 PSTN tab

Následující záložku PSTN nastavíme pro příchozí i odchozí volání. Jednotlivé parametry nastavení zde budou vypsány.

SIP Settings			
SIP Port:	5061	SIP 100REL Enable:	no
EXT SIP Port:		Auth Resync-Reboot:	yes
SIP Proxy-Require:		SIP Remote-Party-ID:	yes
SIP GUID:	no	SIP Debug Option:	none
RTP Log Intvl:	0	Restrict Source IP:	no
Referor Bye Delay:	4	Refer Target Bye Delay:	0
Referee Bye Delay:	0	Refer-To Target Contact:	no
Sticky 183:	no		
Proxy and Registration			
Proxy:	62.44.19.70		
Outbound Proxy:			
Use Outbound Proxy:	no	Use OB Proxy In Dialog:	yes
Register:	no	Make Call Without Reg.:	yes
Register Expires:	3600	Ans Call Without Reg.:	yes
Use DNS SRV:	no	DNS SRV Auto Prefix:	no
Proxy Fallback Intvl:	3600	Proxy Redundancy Method:	Normal
Subscriber Information			
Display Name:		User ID:	O2_pstn
Password:	*****	Use Auth ID:	no
Auth ID:			
Mini Certificate:			

Obr. 20: PSTN záložka - 1.část

- **SIP port: 5061** (pro komunikaci SPA s PSTN)
- **Proxy:** opět **IP adresa** serveru
- **Register: NO**
- **Make Call Without Reg.: YES**
- **Ans. Call Without Reg.: YES**
- **User ID:** údaj, který se použije v *sip.conf* souboru Asterisku
- **Password:** heslo (použité v *sip.conf*)

The screenshot displays the Asterisk configuration interface for VoIP settings. It is divided into three main sections:

- Dial Plans:** A list of eight dial plans. Plan 1 is '(s0<s@62.44.19.70:5060>)', Plan 2 is '(xx.)', Plan 3 is '(2[2-9]xxxxxxxxs0)', Plan 4 is '(xx.)', Plan 5 is '(xx.)', Plan 6 is '(xx.)', Plan 7 is '(xx.)', and Plan 8 is 'S0<:123@62.44.19.70>'. Plans 2 and 8 are highlighted with red boxes.
- VoIP-To-PSTN Gateway Setup:** A series of configuration options. 'VoIP-To-PSTN Gateway Enable' is set to 'yes'. 'VoIP Caller Auth Method' is 'HTTP Digest'. 'VoIP PIN Max Retry' is '3'. 'One Stage Dialing' is 'yes'. 'Line 1 VoIP Caller DP' is '2'. 'Line 1 Fallback DP' is 'none'. 'VoIP Caller Default DP' is 'none'. Below these are fields for VoIP Caller 1 through 8 PINs and their corresponding DPs, all set to '1'.
- VoIP Users and Passwords (HTTP Authentication):** 'VoIP User 1 Auth ID' is 'O2_pstn'. 'VoIP User 1 DP' is '2'. 'VoIP User 1 Password' is masked with asterisks.

Obr. 21: PSTN záložka - 2.část

Zde jsem využíval pouze 2 dial plány, dial plán č. 2 a č. 8. Dial plán 2 umožňuje přepojit jakákoli čísla, která přicházejí na Linku 1. Dial plán č. 8 je pro čísla PSTN: jak je vidět, mám nastaveno, že číslo je převedeno do „uživatele“ 123. To je pro lepší konfigurovatelnost dial plánu v Asterisku. Dále je potřeba nastavit:

- **VoIP Caller Auth Method: HTTP Digest**
- **One Stage Dialing: YES**
- **Line 1 VoIP Caller DP: 2** (dial plan č.2)
- **VoIP Caller Default DP: NONE**
- **VoIP User 1 Auth ID: username** (do přihlašování v *sip.conf*)
- **VoIP User 1 DP: 2**
- **VoIP User 1 Password: heslo** (opět pro *sip.conf*)

PSTN-To-VoIP Gateway Setup			
PSTN-To-VoIP Gateway Enable:	yes	PSTN Caller Auth Method:	none
PSTN Ring Thru Line 1:	no	PSTN PIN Max Retry:	3
PSTN CID For VoIP CID:	yes	PSTN CID Number Prefix:	
PSTN Caller Default DP:	8	Off Hook While Calling VoIP:	no
Line 1 Signal Hook Flash To PSTN:	Disabled	PSTN CID Name Prefix:	
PSTN Caller ID Pattern:			

Obr. 22: PSTN záložka - 3.část

Jak bylo výše uvedeno, nastavení defaultního dial plánu pro PSTN (č.8):

FXO Timer Values (sec)			
VoIP Answer Delay:	0	VoIP PIN Digit Timeout:	10
PSTN Answer Delay:	3	PSTN PIN Digit Timeout:	10
PSTN-To-VoIP Call Max Dur:	0	PSTN Ring Thru Delay:	3
VoIP-To-PSTN Call Max Dur:	0	PSTN Ring Thru CWT Delay:	3
VoIP DLG Refresh Intvl:	0	PSTN Ring Timeout:	4
PSTN Dialing Delay:	1	PSTN Dial Digit Len:	.1/.1
PSTN Hook Flash Len:	.1		

PSTN Disconnect Detection			
Detect CPC:	yes	Detect Polarity Reversal:	yes
Detect PSTN Long Silence:	no	Detect VoIP Long Silence:	no
PSTN Long Silence Duration:	30	VoIP Long Silence Duration:	30
PSTN Silence Threshold:	medium	Min CPC Duration:	0.5
Detect Disconnect Tone:	yes		
Disconnect Tone:	440@-30,425@-30;5(.165/.165/1+2)		

International Control			
FXO Port Impedance:	220+820 120nF	Ring Frequency Min:	10
SPA To PSTN Gain:	0	Ring Frequency Max:	100
PSTN To SPA Gain:	0	Ring Validation Time:	256 ms
Tip/Ring Voltage Adjust:	3.5 V	Ring Indication Delay:	512 ms
Operational Loop Current Min:	10 mA	Ring Timeout:	640 ms
On-Hook Speed:	Less than 0.5 ms	Ring Threshold:	13.5-16.5 Vrms
Current Limiting Enable:	no	Ringer Impedance:	High (Normal)
Line-In-Use Voltage:	30		

Obr. 23: PSTN záložka - 4.část

Hodnoty, které jsou zde uvedeny, jsem použil díky návodu Trixbox Without Tiers z následující stránky[10]. Hodnota Disconnect Tone je tón pro Českou republiku[38].

6.2.4 User 1 tab

Zde je potřeba nastavit tyto hodnoty.

- **Default Ring: 3**
- **Default CWT: 8**

Po provedení všech změn je nutné kliknout na tlačítko **Submit All Changes** na konci stránky ve webovém rozhraní. Tím se změny zapíší do SPA.

6.3 Konfigurační soubor SIP.CONF

V konfiguračním souboru *sip.conf* jsou definováni uživatelé pro přístup do Asterisku, odkazy contextů pro přístup do *extensions.conf* souboru, což je hlavní soubor pro vytváření dial plánů, který umožňuje danému uživateli provést odchozí hovor.

Hlavní sekce *general* nám umožňuje definovat základní parametry Asterisk ústředny.

```
[general]
bindport=5060                ;port pro SIP přenosy
bindaddr=0.0.0.0            ;IP adresa pro navázání spojení (0.0.0.0
                             pro všechny adresy)
disallow=all                 ;Zakázání všech kodeků
allow=ulaw                   ;povolení vybraných kodeků
allow=alaw
allow=gsm

register => 511119595:heslo@212.71.146.175:5060/511119595
                             ;register umožňuje stálé přihlášení pro
                             dané číslo. V tomto případě je to pro VoIP
                             operátora 802.cz
```

Následuje hlavní část souboru *sip.conf*, kde se přidávají jednotliví uživatelé. V následující části se podíváme na jednotlivé služby z SPA (jak bylo vyplněno ve webovém rozhraní – User ID a hesla).

```
[517350051]                  ;název pro pevnou linku (čís. pevné)
type=friend                  ;friend typ určuje že daný uživatel
                             může poskytovat příchozí i odchozí
                             hovory

fromuser=02_pstn
username=02_pstn             ;User ID
secret=heslo                 ;heslo
host=192.168.1.2             ; IP adresa SPA3102
port=5061                    ;port pro přenosy PSTN
dtmfmode=rfc2833
insecure=port,invite        ;určuje jak se má nakládat s
                             připojením daného uživatele
context=main                 ;určuje hlavní kontext v dialplanu,
                             který má tento uživatel povolen

[spa]                        ;definování link1 (Line 1 v SPA)
type=friend
host=192.168.1.2
context=main
username=line1
userid=line1 <123>
secret=heslo
dtmfmode=rfc2833
disallow=all
allow=ulaw

[802_VoIP]                   ;VoIP operátor
type=friend
host=hlas.802.cz            ;definování hostname 802.cz
```

```
username=511119595 ;tel. čís. reg. služby na 802.cz
fromuser=511119595
secret=heslo
fromdomain=hlas.802.cz
dtmfmode=rfc2833
allow=alaw
allow=ulaw
permit=212.71.146.175 ;povolení IP adresy
insecure=invite
context=main
```

Používané kontexty vyjadřují pro daného uživatele, který kontext se má použít v hlavním souboru dial planu (*extensions.conf*). Kontext nám odkazuje na to, co se má provádět při odchozím hovoru od daného uživatele. Uvedu zde 2 uživatele, které mám zapsány v *sip.conf*. Ostatní jsou zapsáni velmi podobně.

```
[222]
type=friend
username=222
userid=test2 <222> ;userid označuje číslo, které se má
                    ;použít pro tohoto uživatele <222>
host=dynamic ;zjistí se aktuální IP adresa z
              ;INVITE zprávy (příchozí volání)
secret=heslo
context=main

[111_echo] ;uživatel 111 pro echotest
type=friend
username=222
userid=echotest <111>
host=dynamic
secret=heslo
```

Jak je výše vidět, uživatel 111 (echotest) nemá definovaný kontext. To je z toho důvodu, že jsem nepočítal, že by se na něj kdokoli přihlašoval. Čili není potřeba odkaz na odchozí hovory z tohoto uživatele. Tento „uživatel“ slouží pouze ke zkoušce, zda-li spojení je funkční. Přehrává se demo Asterisku (tzv. *Welcome demo*), kde lze i vyzkoušet, zda je správně funkční mikrofón a zda-li je nějaké latence v hovoru či nikoli.

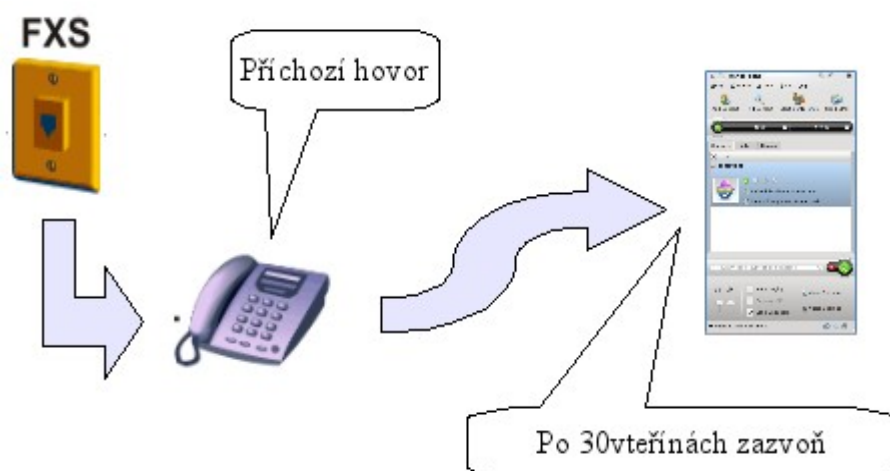
Zajisté existuje plno příkazů a možností, jak tento soubor doplnit a obohatit o funkce Asterisku, ale vzhledem k daným plánům není více parametrů potřeba. Pokud tak někdo chce učinit a hledá informace, může tak učinit zde[39]. V následující podkapitole objasním, proč tak činím.

6.4 Návrh dial plánu

Předtím, než začneme něco konfigurovat, je dobré si ujasnit jaká byla moje hlavní myšlenka propojení SPA3102 brány, Asterisku a PSTN linky k volání na jednotlivé poskytovatele a na volání na mou ústřednu z vnějších sítí.

6.4.1 PSTN síť – PSTN číslo O2 (příchozí a odchozí hovory)

Příchozí hovor vypadá následovně. Jelikož máme u nás doma rozdvojenou O2 linku (čili v jedné je napojen telefon natrvalo k PSTN, a druhá jde do mé SPA brány), tak pro mne byla hlavní myšlenka ta, že by příchozí hovor zvonil nejprve na telefonu připojeném přímo do PSTN a po nějakém čase, dejme tomu 30vteřin, by byl hovor přeměrován do mého počítače na účet v Asterisku.



Obr. 24: Příchozí hovor - PSTN (návrh)

V extensions.conf souboru by tato žádost mohla vypadat následovně:

```
[main]
exten => 123,1,NoOP(${CALLERID})      ;ukáže pouze aktuálně volané číslo
exten => 123,n,Answer()                ;zvedne sluchátko
exten => 123,n,Ringing()               ;zvonění na lince
exten => 123,n,Wait(30)                ;čekací doba ve vteřinách
exten => 123,n,Dial(SIP/222)           ;po uplynutí času odkazuj na 222
                                         uživatel
exten => 123,n,Hangup()                ;po ukončení zavěsí sluchátko
```

Takto je nakonfigurován příchozí hovor na pevnou linku. **Odchozí hovor** (čili volání skrz PSTN síť) jsem si navrhl následovně. Jelikož v SPA v záložce Line 1 mám uveden dial plán, který umožňuje volání skrz PSTN, díky použití prefixu #, mohl bych následující pravidlo vypustit, ale neudělám to, a to z důvodu, že se mi zdá, že při vytáčení díky nastavení volání na PSTN skrz Asterisk, je vytáčení rychlejší, než při vytáčení přes

SPA bránu.

Proto byl nastaven prefix, přes který se volá skrz PSTN linku. Nastavil jsem si číslo 1, volání pak probíhá tak, že volající vytočí PSTN číslo následovně:

1212345678 = **prefix 1+212345678** (vytáčené číslo).

```
exten => _1XXXXXXXXXX,1,NoOp ;výpis čísla
exten => _1XXXXXXXXXX,n,Dial(SIP/${EXTEN:1}@517350051,30,r)
exten => _1XXXXXXXXXX,n,Hangup ;položení sluchátka
```

Druhý řádek nám říká, že pro vytočení přes prefix 1 se má použít odchozí volání z uživatele 517350051 ze souboru *sip.conf*, číslo se ořeže o první znak (EXTEN:1) a, délka vyzvánění bude omezena na max.30 vteřin.

Pro PSTN jsou tedy 2 volby jak vytáčet číslo:

- #212345678
- 1212345678

6.4.2 VoIP operátor (příchozí a odchozí hovory)

Pro využití VoIP volání mám zaregistrován účet u 802.cz, který je dle mého osobního průzkumu asi nejlevnější operátor v České republice. **Příchozí hovory** mám v dial plánu mám řešeny následovně:

```
exten => 511119595,1,Dial(SIP/222,20)
exten => 511119595,n,Playback(vm-nobodyavail)
```

První řádek stanoví, že příchozí hovor je přeměrován rovnou na uživatele 222 a po 20ti vteřinách vyzvánění je přehrána volajícímu hláška, že nikdo není momentálně přítomen, poté se hovor ukončí. **Odchozí hovor** je řešen vytočením standardního čísla bez mezinárodní předvolby:

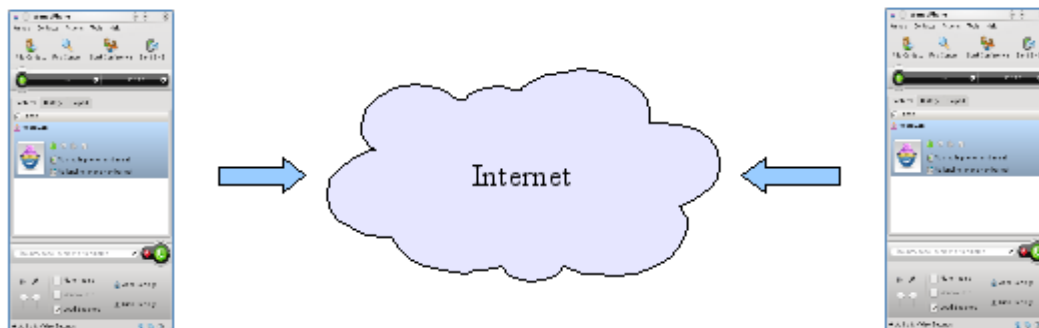
```
exten => _XXXXXXXXXX,n,Answer()
exten => _XXXXXXXXXX,n,Ringing()
exten => _XXXXXXXXXX,n,Dial(SIP/${EXTEN:0}@802_VoIP,,r)
exten => _XXXXXXXXXX,n,Hangup
```

Význam třetího řádku: Pro vytáčení použij uživatele 802_VoIP (ze souboru *sip.conf*), volané číslo vytáče celé, jak je zadáno (EXTEN:0), zvoň dokud nezavěším nebo dokud protistrana nevezme telefon.

6.4.3 Volání v rámci lokální sítě (příchozí i odchozí hovory)

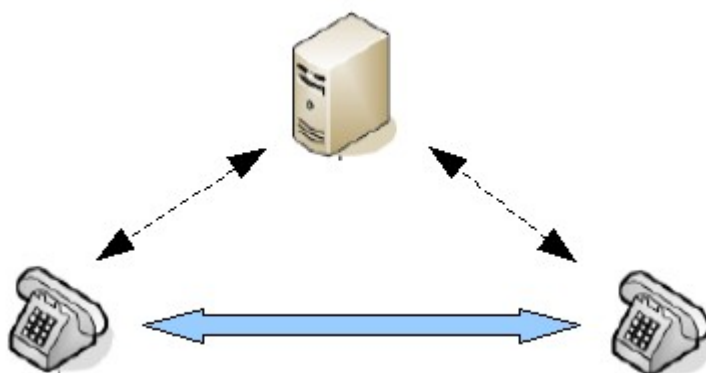
Tento způsob se dá velmi dobře využít i na standardní volání, a to zdarma! Pokud se povede přihlášení a zavolání si mezi klienty přes internet, je možné toto provádět i jako

službu podobnou Skype. Stačí jen přidat uživatele do skupiny a poslat mu přihlašovací údaje. Druhá strana se připojí na jakýkoli SW telefon a pokusí se o přenos hlasu po internetu.



Obr. 25: Volání v rámci internetu

V rámci mé sítě mi toto volání nefungovalo, jen při zavolání mezi telefonem (napevno nastavené číslo) a přihlášeným uživatelem v Linuxu přes WengoPhone. Bohužel volání mezi notebookem a počítačem nefungovalo. I když jsem povolil potřebný port 5060 jak na notebooku (v shorewall), tak i na počítači a na SPA jsem dal přenášet pakety na tomto portu, nenastalo nic a samotný WengoPhone se ani nepřihlásil k serveru. Ale tuto možnost jsem zkusil s kamarádem, připojeným ve vnějším internetu, a po několika zklamáních jsme si nakonec zavolali. Čili služba funguje, ale záleží na tom, jaké má protistrana připojení nebo zda-li má počítač za routerem s NATem.



Obr. 26: Volání v rámci vnitřní sítě

V dial plánu to bude vypadat následovně:

```
[main]
exten => _XXX,1,Dial(SIP/${EXTEN})

[inside]
exten => _XXX,1,Dial(SIP/${EXTEN},10)
exten => _XXX,n,Goto(s-${DIALSTATUS},1) ;Skok na NOANSWER odpovědi
(uzivatel není dostupný)
exten => s-NOANSWER,1,Playback(vm-nobodyavail)
exten => s-NOANSWER,2,Wait(1)
exten => s-NOANSWER,3,Hangup()
```

V main větvi je definováno vytváření extensions pro volající čísla. Kontext [inside] obsahuje plán na příchozí volání. První řádek jako tradičně obsahuje čekání 10 vteřin, pak se provede další příkaz a ten nám stanoví, že pokud volaný nevykazuje žádnou odpověď, přejde se na další řádek, kde se přehraje, že je dotyčný nepřítomen. Pak se počká 1 vteřinu a hovor se ukončí.

Celý *extensions.conf* společně se souborem *sip.conf* lze nalézt v příloze, kde je vše pohromadě.

7 ENUM

V této diplomové práci mám řešit i ENUM technologii. Jak již bylo napsáno, ENUM umožňuje zdarma volat na běžná telefonní čísla v libovolné síti pomocí internetu a výrazně tak ušetřit finanční prostředky díky namapování telefonních čísel do DNS. Tím pádem se vede číslo přímo po internetu a ne přes obvyčejného poskytovatele.

Podmínky volání s ENUM jsou tyto:

- Volající musí podporovat ENUM
- Příjemce má příchozí SIP číslo
- Příjemce nadefinuje ENUM záznam se SIP číslem
- Volající položí dotaz do DNS
- Pokud se najde odpovídající ENUM záznam, bude použit

Zprovoznění na mé PBX vypadá asi nějak následovně: pro vyhledávání ze záznamů e164.arpa bylo potřeba vytvořit makro pro ENUM, které to umožňuje. Jelikož hlavní poskytovatel ENUM domén u nás v České republice www.nic.cz nabízí přímo část tohoto macra (převzaté z dokumentace o ENUM) na svých stránkách[40], nebyl žádný problém jej implementovat do mého *extensions.conf* souboru.

```
[macro-enum]
;ARG1 je hledane cislo, ARG2 je schema (sip, iax2)
exten => s,1,Set(sipcount=${ENUMLOOKUP(+${ARG1},${ARG2},c)}|counter=0)
exten => s,n,GotoIf("${counter}"<"${sipcount}")?start:end)
exten => s,n(start),Set(counter=${counter}+1)
exten => s,n,Dial(SIP/${ENUMLOOKUP(+${ARG1},${ARG2},${counter}))
exten => s,n,GotoIf("${counter}"<"${sipcount}")?start:end)
exten => s,n(end),NoOp("Zadne validni ${ARG2} zaznamy v e164.arpa pro ${ARG1}")
```

Pro odchozí hovory je použita tato část konfiguračního souboru, která je v [main] části:

```
; ENUM pro mezinarodni cisla
; je zapotrebi odstranit dve nuly na zacatku
exten => _00X.,1,Macro(enum,${EXTEN:2},sip)
exten => _00X.,n,NoOp("MEZINARODNI FORMAT - OREZ 00")
exten => _00X.,n,Macro(enum,${EXTEN:2},iax2)
exten => _00X.,n,Dial(${OUTBOUNDTRUNK}/${EXTEN})
; ENUM pro ceska cisla
exten => _X.,1,Macro(enum,${EXTEN:0},sip)
exten => _X.,n,NoOp("CESKY FORMAT - NIC NEOREZE")
exten => _X.,n,Macro(enum,${EXTEN:0},iax2)
; ostatni - volame pres PSTN
exten => _X.,n,Dial(${OUTBOUNDTRUNK}/${EXTEN})
```

Celá funkce ENUMu a obyčejného vytáčení funguje takto: pokud chceme volat a zároveň zjistit, zda volaný účastník má registrované ENUM číslo, je potřeba jej vytáčet v těchto tvarech:

- +420XXXXXXXXXX
- 00420XXXXXXXXXX
- 420XXXXXXXXXX

Tímto se budou automaticky kontrolovat záznamy na e164.arpa a pokud bude odpověď kladná, tak se tímto způsobem propojí oba volající a volání zdarma je poskytováno. Pokud však dané číslo nebude nalezeno, bude se postupovat standardním vytáčením přes VoIP poskytovatele (802.cz) a hovor bude tarifikován podle tohoto providera. Když bude vytočeno číslo standardním způsobem, bez mezinárodní předvolby pro Českou republiku, tak bude hovor směřován přímo na VoIP operátora.

Na zkoušku, zda-li voláme přes standardního operátora nebo přímo přes ENUM, vytvořilo sdružení www.nic.cz 3 demo telefonní čísla, na kterých je možnost si testovat funkčnost ENUMu. IVR nám odpoví, z jakého účtu voláme a pokud řekne „PSTN“, tak voláme standardní cestou a je zde tarifikace podle našeho providera nebo řekne ENUM, pak je vše nastaveno v pořádku a volání zdarma je poskytováno. Lze se o tom přesvědčit také např. v admin sekci svého VoIP providera, kde by hovor neměl být vidět v odchozích hovorech.

Čísla pro test ENUM jsou tyto:

- +420222745120 – poslech muziky s informací PSTN nebo ENUM
- +420222745121 – echotest Asterisku pro zjištění latence na lince
- +420222745122 – demo využívající DTMF tónů

8 WWW ROZHRAŇÍ PRO SEBEREGISTRACI NOVÝCH UŽIVATELŮ

Jelikož původní zadání nemohlo být dokončeno a vzájemná integrace mezi OpenSERem a Asteriskem se nezdařila, rozhodl jsem se tedy tento server navrhnout pro domácí využití s aplikacemi hovorů z mé ústředny a do mé ústředny za pomoci SPA3102 brány, která poskytuje připojení PSTN sítě a umožňuje přes něj volat, tak i pomocí VoIP providera a také implementaci a registraci ENUM domény, jak bylo původně v zadání.

Proto pro mne, jako obyčejného uživatele, který využívá svoji osobní pobočkovou ústřednu k vlastním a případně i rodinným účelům, není potřeba řešit původní zadání, a to bod číslo 3, kde je potřeba navrhnout a zrealizovat www rozhraní pro seberegistraci nových uživatelů. Jelikož toto lze pro osobní účely provádět samostatně a v rámci administrace serveru velmi účelně, nebyl tento bod vypracován. Z hlediska pobočkové ústředny, která slouží převážně k takovýmto osobním účelům je to vyhovující.

ZÁVĚR

V této diplomové práci nastalo pár neočekávaných situací, které mi ztížily její dokončení. Předně, návrh zadání, které mělo vyhovovat pro pobočkovou ústřednu pro středně velké firmy s velkou zátěží, se po čase projevil jako celkem velký oříšek. Provádět takto značnou tvorbu serveru doma bylo velmi obtížné. Nejdříve nastal menší problém v mém připojení, bylo nutné pořídit si veřejnou IP adresu.

Další problém nastal s posíláním paketů na port UDP a celkové problémy s povolením jen portu UDP. Pro přístup jsem měl povolen port 22 SSH a UDP 5060 na mém wifi-routeru, který jsem využíval pro své účely a připojení notebooku. Ukázalo se, že na tento router být server připojen nemůže. Veřejná adresa, která byla nastavena v routeru a následné přeposílání paketů na portu UDP 5060 na server vedly k tomu, že jsem musel router vyjmout z mé sítě a server napojit přímo do internetu, aby měly pakety, které k mému serveru proudily, volnou cestu. Router tyto pakety někde pohlcoval, bohužel jsem nepřišel na to, kde k tomu přímo docházelo. Dále bylo potřeba vyřešit na serveru firewall a povolení portů, zde už vše fungovalo jak mělo a OpenSER fungoval pouze na UDP portu. Největší problém měl však teprve přijít.

Pro realizaci této práce bylo potřeba, aby servery OpenSER a Asterisk byly vzájemně propojeny, využívaly jedné databáze a volání přes Asterisk mělo vést k používání uživatelů, kteří se registrovali do databáze přidáním pomocí OpenSERu. Bohužel se tak nestalo, strávil jsem mnoho času řešením tohoto problému, který nevedl ke zdárnému konci. Uživatelé připojení díky portu 5065 na Asterisk si vzájemně zavolali, ono si vlastně zavolali i uživatelé z OpenSERu, ale problém vznikl v tom, že jsem si přidal např. určité pravidlo do dialplanu, „zvoň 10vteřin a poté, když nikdo nezvedne telefon přehraj hlášku, že volaný účastník není dostupný“. Toto nebyl problém pro uživatele Asterisku, u nich vše probíhalo v naprostém pořádku, ale u uživatelů přihlášených přes OpenSER (port 5060) se přesměrování na port Asterisku nekonalo a uživatelům zvonil telefon neustále dokola. Při hledání řešení jsem narazil na jednu věc a to tu, že občas bývá problém tyto dva programy spojit, aby fungovaly jako celek. I přes různé přenastavování konfiguračních souborů a zkoušek se mi tohoto cíle nepodařilo dosáhnout. Po konzultaci s mým vedoucím mi bylo navrženo dokončení této diplomové práce jen s pomocí Asterisku. A zde jsem to pojal podle své myšlenky vytvořit si svoji osobní pobočkovou ústřednu, která bude vyhovovat jak zadání diplomové práce, tak zároveň i mé osobní potřebě. Ústředna by měla poskytovat hovory do jakékoli sítě, jak mobilní tak i pevné, a to současně při využití co nejlevnějšího volání. Aby byla možnost se i na server dovolat

z jakéhokoliv čísla. Vše propojit a využívat to vzájemně s SPA3102 bránou a VoIP providerem.

I když v zadání této diplomové práce bylo provést pouze hovory do ostatních sítí, vyřešil jsem také příchozí hovory, což jsem bral jako samozřejmost. Jednotlivé volání je popsáno v teoretické části, zde jen okrajově. Server dokáže volat díky SPA3102 do pevné sítě, a to přes standardní prefix #, který byl navržen v dialplanu SPA od 802.cz, dále přes mnou definovaný prefix 1 v Asterisk dialplanu, jak jsem zmínil, možná se jedná o rychlejší vytáčení (subjektivní názor). Dále při běžném vytáčení čísla se provede volání přes VoIP operátora 802.cz, lze zadat jakýkoli formát. Při vytáčení telefonního čísla v tzv. mezinárodním formátu se zjistí, zda-li číslo není registrované pod ENUM doménou a neumožňuje tak spojit telefonní číslo přes internet díky záznamům v e164.arpa. Poté je řešeno i volání uživatelů po internetu, po přidání na server Asterisku a jejich následné přihlášení, např. z domova na softwarovém telefonu. Příchozí hovory jsou u pevné linky řešeny tím způsobem, že telefon, který je neustále připojen do PSTN sítě, bude zvonit nějaký časový interval a poté se hovor přeměruje na uživatele přihlášeného na počítači a zde bude probíhat jeho následné zvonění. Řešení ohledně volání na číslo, které mám registrováno u VoIP providera, bylo nastaveno na volání přímo do počítače na přihlášeného uživatele. Bylo nastaveno i pár upozornění na to, že volaný účastník není dostupný, ale nakonec jsem tyto IVR hlášky odstranil, jelikož to nakonec stejně pro volajícího znamenalo ztrátu peněz za propojení hovoru, a proto jsem tyto hlášky použil jen u volání v rámci IP telefonie, čili uživatel mezi uživatelem (podobně jako Skype).

Uvědomuji si, že by bylo možné provádět nepřeberné množství nastavení a přidání funkcí pro tuto pobočkovou ústřednu, ale v rámci této práce na ně nebyl brán zřetel. Funkci plní v maximálním rozsahu.

Z toho vyplývá má úvaha nad posledním bodem v zadání mé práce, seberegistraci uživatelů pro server v rámci www rozhraní. Jelikož jsem PBX nakonec díky nefunkčnosti realtime integrace OpenSERu a Asterisku přehodnotil, není potřeba jej v rámci této diplomové práce provádět, neboť registrace lze provádět přímo na serveru. Z toho tedy vyplývá, že PBX nebude sloužit v prostředí s vytížením tisícičkami uživatelů.

Jak již bylo napsáno, pro domácí využití klasické pevné linky, moderního volání přes levné VoIP operátory a volání mezi uživateli je tato domácí PBX nadměru uspokojivá. Rád jsem pracoval na této práci, jelikož jsem se dozvěděl mnoho informací a zajímavostí, které by mi jinak ušly.

ZÁVĚR V ANGLIČTINĚ

There occurred a few unexpected situations while I was working on this thesis, which made it harder to complete. First, the draft of the task setting, which should have complied with the swichboard for midsize businesses, with a high load, showed, in the course of time, as a real problem. Trying to implement the creation of such a large server at home was very difficult. For a start, there was a smaller problem in my connection, it was necessary to get a public IP address.

Another problem occurred when I tried sending the packets on UDP port and there were general problems with the authorization of the UDP port only. I was allowed to access to SSH port 22 and UDP 5060 on my wifi-router, which I use for personal purposes and for connecting my laptop. It became clear that the server cannot be connected to this router. The public address, which was set in the router and the subsequent forwarding of packets on UDP port 5060 to the server meant that I had to remove the router from my network and connect the server directly to the Internet, so that packets could freely flow to my server. The router had absorbed these packets somewhere; I was not able to find out where exactly this was happening. Furthermore, it was necessary to solve the firewall on the server and the ports authorization; luckily, everything worked as it should, and OpenSER worked only on UDP port. The biggest problem, however, was yet to come.

To be able to carry out this work I needed OpenSER and Asterisk to be interconnected, to use one database, and calls through Asterisk should lead to the use of users who register in the database using the added OpenSER. Unfortunately, this did not happen. I spent much time addressing this problem but this did not lead to a successful outcome. Users connected through port 5065 to Asterisk could call one another, even the users connected through OpenSER could call, but a problem arose when I added a rule to the dialplan, i.e. "Ring for 10 seconds and if nobody picks up the phone play the message saying that the the person is not available. " This was not a problem for users of Asterisk, everything was in perfect order with them, but users registered via OpenSER (port 5060) were not redirected to port Asterisk and the phone kept ringing. When searching for a solution I have found that sometimes it is a problem to combine these two programs to work as a whole. Despite trying to set various configuration files and tests I failed to achieve this goal. Having consulted the problem with my supervisor, I was to complete the thesis with the help of Asterisk only. And at this moment I decided to work independently. I wanted to create my private swichboard, which will satisfy both the thesis task setting and also my personal needs. The exchange should provide calls to any network, both mobile

and public switched telephone network, and at the same time, using the cheapest call. Also, it should be possible to call the server from any number. Connect it all and use it with the SPA3102 gateway and VoIP provider.

Although the task setting of this thesis was to make only calls to other networks, I also resolved incoming calls, because I took them for granted. Each call is described in detail in the theoretical part, only marginally here. The server can call thanks to the SPA3102 to the public switched telephone network via a standard prefix #, which was designed in dial planu SPA from 802.cz, then through a defined (I did this) prefix 1 in Asterisk dial plan. As I mentioned, maybe this is a faster dial-up (subjective view) . Moreover, the the call is carried out via a VoIP call 802.cz operator when dialing numbers normally but you can specify any format. When dialing a phone number in the international format, it will be found out whether a number is not registered under the ENUM domain and does not allow to connect a phone number via the Internet through the records in e164.arpa. Then users' calling through the Internet is addressed after they join to the Asterisk server and then log on, for example, from home on the phone software. Incoming calls in landlines are handled in the following way: the phone is constantly connected to the PSTN network and it will ring for some time and then the call is redirected to the user logged on the computer and the ringing will be heard there. The solution for the call to a number that I registered with a VoIP provider, had been set to call directly into the computer of the logged-on user. A couple of warnings had also been set to say that the called party is not available but I removed the IVR messages in the end because this would ultimately mean spending money for the caller as this was a standard link and, therefore, I used these warning messages only for calls within the IP telephony, or the user between the user (like Skype).

I realize that it would be possible to perform a great number of various other settings for this switchboard but in the context of this work they were not taken into account. The required function is fully working.

The last point in the task setting of my work was self-registration of users on the server within the web interface. Since I finally reviewed using PBX due to malfunction of relatime OpenSER and Asterisk integration, there is no need to carry it out in the context of this thesis because the registration can be done directly on the server. It therefore follows that the PBX will not be used for high load by thousands of users.

As suggested, for home use of traditional land lines, a modern low-cost VoIP calls through operators, and calls between users, the home PBX is highly satisfactory. I was glad

to work on this task because I learned a lot of information and facts which I would not know otherwise.

SEZNAM POUŽITÉ LITERATURY

- [1] Informace o telefonním systému IP PBX/VoIP/SIP [online]. [cit. 2008-1-23].
Dostupný z WWW: <<http://www.3cx.cz/voip-sip/ip-pbx-faq.php>>
- [2] SIP [online]. [cit. 2008-1-23]. Dostupný z WWW: <<http://www.voip-info.org/wiki-SIP>>
- [3] Pomoc, návody a často kladené dotazy o SIP IP telefonii [online]. [cit. 2008-1-22].
Dostupný z WWW: <<https://sip.osu.cz/hlp/index.php>>
- [4] ENUM – Nové horizonty [online]. [cit. 2008-1-22]. Dostupný z WWW:
<http://www.iptw.cz/files/ENUM-Nove_horizonty.pdf>
- [5] ENUM – systém pro veřejnou komunikaci přes protokol SIP [online]. [cit. 2008-1-20]. Dostupný z WWW:
<<http://www.computerworld.cz/cw.nsf/id/7CA63B75AFF0219AC12571E2003CE1A0?OpenDocument&cast=1>>
- [6] Signalizace SIP protokolu [online] [cit.2009 – 3 – 1]. Dostupný z WWW:
<http://www.ip-telefon.cz/archiv/dok_osta/ipt-2006_Signalizace_SIP.pdf>
- [7] Asterisk a jeho použití [online] [cit. 2009 – 4 – 1]. Dostupný z WWW:
<http://www.cesnet.cz/doc/seminare/20051115/pr/voz05_asterisk.pdf>
- [8] Asterisk PBX [online] [cit. 2009 – 4 – 24]. Dostupný z WWW:
<<https://sip.cesnet.cz/cs/swahw/asterisk>>
- [9] ENUM [online] [cit. 2009 – 4 – 15]. Dostupný z WWW:
<<https://sip.cesnet.cz/cs/protokoly/enum>>
- [10] Trixbox without Tears [online] [cit. 2009 – 3 – 6]. Dostupný z WWW:
<http://dumbme.mbit.com.au/trixbox/trixbox_without_tears.pdf>
- [11] Asterisk VoIP – ústředna [online] [cit. 2009 – 4 – 13]. Dostupný z WWW:
<<http://www.abclinuxu.cz/clanky/site/asterisk-voip-ustredna-1-planovani>>
- [12] Asterisk The Future of Telephony [online] [cit.2009 – 5 – 10]. Dostupný z WWW:
<<http://downloads.oreilly.com/books/9780596510480.pdf>>
- [13] ENUM konec minutových poplatků [online] [cit 2009 – 3 – 15]. Dostupný z WWW:
<<http://www.zive.cz/Clanky/ENUM--konec-minutovych-poplatku/sc-3-a-138560/default.aspx>>

- [14] VoIP-INFO - wiki[online] [cit 2009 – 2 – 12]. Dostupný z WWW:
<<http://www.voip-info.org/wiki/>>
- [15] OpenSIPS project – dokumentace [online] [cit.2009 – 3 – 22]. Dostupný z WWW:
<<http://www.opensips.org/Resources/Documentation>>
- [16] České fórum zabývající se IP telefonii [online] [cit.2009 – 3 - 11]. Dostupný z
WWW: <<http://www.telefonujeme.cz/>>
- [17] Wikipedia CS [online] [cit.2009 – 3 – 25]. Dostupný z WWW:
<<http://cs.wikipedia.org>>
- [18] Instalace OpenSER [online] [cit.2009 – 4 – 1]. Dostupný z WWW:
<http://econf.switch.ch/econfportal/www/page_viewer?id=/documentation/ecs/technical_infos/openser.html>
- [19] IP telefonie [online] [cit.2009 – 3 – 22]. Dostupný z WWW: <<http://www.cesnet.cz/iptelefonie/>>
- [20] SER – Getting started příručka [online] [cit.2009 – 3 – 15]. Dostupný z WWW:
<<http://siprouter.teigre.com/doc/SER-GettingStarted.pdf>>
- [21] Iptel.org [online] [cit.2009 – 3 – 20]. Dostupné z WWW: <<http://www.iptel.org/>>
- [22] Getting started with OpenSER [online] [cit.2009 – 4 – 4]. Dostupný z WWW:
<<http://consciencspeaks.blogspot.com/2008/06/getting-started-with-openser.html>>
- [23] ENUM nic.cz [online] [cit.2009 – 4 – 11]. Dostupný z WWW:<<http://enum.nic.cz/>>
- [24] VoIP projekt [online] [cit.2009 – 3 – 20]. Dostupný z WWW: <http://homel.vsb.cz/~voz29/team_2007-VoIP-Po9.00/index.html>
- [25] Začátečnická příručka OpenSER [online] [cit.2009 – 3 – 11]. Dostupný z WWW:
<http://www.oldskoolphreak.com/tfiles/voip/beginners_openser.txt>
- [26] Kamilio [online] [cit.2009 – 3 – 13]. Dostupný z WWW:
<<http://www.kamilio.org/>>
- [27] Srovnání OpenSER a Asterisk [online] [cit.2009 – 3 – 25]. Dostupný z WWW:
<<http://www.packtpub.com/article/comparing-asterisk-and-openser>>
- [28] Realtime integrace OpenSER a Asterisk [online] [cit.2009 – 3 – 10]. Dostupný z
WWW: <<http://www.voip-info.org/wiki/view/Realtime+Integration+Of+Asterisk+1.4+With+OpenSER+1.3>>
- [29] Registrace ENUM domény [online] [cit.2009 – 5 – 15]. Dostupný z WWW: <<http://>

www.voipex.cz/>

- [30] VoIP provider 802.cz [online] [cit.2009 – 5 – 15]. Dostupný z WWW: <<https://www.802.cz/>>
- [31] Konfigurace SPA3102 [online] [cit.2009 – 5 – 10]. Dostupný z WWW: <<http://forum.voxilla.com/linksys-sipura-voip-support-forum/starter-spa3102-asterisk-setup-18612.html>>
- [32] Návod pro konfiguraci Asterisk ústředny [online] [cit.2009 – 5 – 18]. Dostupný z WWW: <<http://www.telegro.cz/>>
- [33] Wiki pro Asterisk [online] [cit.2009 – 5 – 15]. Dostupný z WWW: <<http://www.asteriskguru.com/>>
- [34] Xlite SW telefon [online] [cit.2009 – 4 – 27]. Dostupný z WWW: <<http://www.counterpath.com/x-lite.html&active=4>>
- [35] WengoPhone sw telefon [online] [cit.2009 – 3 – 2]. Dostupný z WWW: <<http://www.qutecom.org/>>
- [36] Zaznamenávání Syslogu SPA3102 [online] [cit.2009 – 5 – 23]. Dostupný z WWW: <<http://www.voip-info.org/wiki/view/Sipura+Syslogging>>
- [37] Co je ENUM? [online] [cit.2009 – 5 – 22]. Dostupný z WWW: <<http://www.swsystems.cz/index.php?hkat=1&kat=46>>
- [38] Nastavení Disconnect Tone [online] [cit.2009 – 4 – 25]. Dostupný z WWW: <http://wiki.telefonujeme.cz/hardware/adaptery/sipura_ata>
- [39] Asterisk sip.conf [online] [cit.2009 – 5 – 11]. Dostupný z WWW: <<http://www.voip-info.org/wiki/view/Asterisk+config+sip.conf>>
- [40] Podpora ENUM nic.cz [online] [cit.2009 – 5 – 22]. Dostupný z WWW: <<http://podpora.nic.cz/page/537/konfigurace-odchozich-hovoru-pres-enum/>>
- [41] Návod na 802.cz pro SPA3102 [online] [cit.2009 – 3 – 13]. Dostupný z WWW: <<https://www.802.cz/navody/navody.php?typ=SPA-3102>>
- [42] Telegro, návody SIP [online] [cit.2009 – 5 – 20]. Dostupný z WWW: <<http://www.telegro.cz/2009/03/17/openser-ser-se-spojuji-vznikne-sip-router>>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

VoIP	Voice over IP, čili volání přes Internet
ENUM	Mapování tel.č. a URI adres přes DNS
DNS	Domain Name Server. Převod domén na IP adresy a opačně.
UDP	Jednodušší přenos dat po Internetu, nevyžaduje takovou režii přenosu jako TCP protokol
RFC	Označování řady standardů a dalších dokumentů popisujících Internetové protokoly.
SIP	Protokol určený pro přenos signalizace v internetové telefonii.
PSTN	Standardní telefonní síť (pevná linka), pro veřejnost k uskutečňování místních, meziměstských a mezinárodních telefonních hovorů.
IVR	Hlasový automat (Interactive Voice Response) používaný např. u operátorů („pro vstup do menu stiskněte 1“)
PBX	Pobočková ústředna
IAX	Protokol určený pro přenos signalizace v internetové telefonii.
H.323	Protokol určený pro přenos signalizace v internetové telefonii.
FXS	Standardní „zdířka“ ve zdi pro pevnou linku
FXO	Port na telefonu.
NAPTR	(Naming Authority PoinTeR) DNS záznam, který obsahuje informace o jednotlivých URI IP službách dostupných pro dané telefonní číslo (doménové jméno)

SEZNAM OBRÁZKŮ

Obr. 1: Inicializace spojení Joe – Bob [6].....	16
Obr. 2: Registrar server [6].....	17
Obr. 3: Redirect server a přesměrování hovorů [6].....	18
Obr. 4: Transakce [21].....	23
Obr. 5: Dialog [21].....	23
Obr. 6: Registrace [21].....	24
Obr. 7: Topologie domácího zapojení.....	34
Obr. 8: X-Lite.....	44
Obr. 9: Nastavení sítě (X-Lite).....	45
Obr. 10: Seznam proxy serverů (X-Lite).....	45
Obr. 11: Nastavení proxy (X-Lite).....	46
Obr. 12: WengoPhone - nastavení účtu.....	47
Obr. 13: Hlavní okno programu WengoPhone.....	48
Obr. 14: Linksys SPA3102.....	55
Obr. 15: Line 1 záložka - 1.část.....	56
Obr. 16: Line 1 záložka - 2.část.....	57
Obr. 17: Line 1 záložka - 3.část.....	57
Obr. 18: Line 1 záložka - 4.část.....	57
Obr. 19: Regional – české oznamovací tóny.....	58
Obr. 20: PSTN záložka - 1.část.....	59
Obr. 21: PSTN záložka - 2.část.....	60
Obr. 22: PSTN záložka - 3.část.....	61
Obr. 23: PSTN záložka - 4.část.....	61
Obr. 24: Příchozí hovor - PSTN (návrh).....	64
Obr. 25: Volání v rámci internetu.....	66
Obr. 26: Volání v rámci vnitřní sítě.....	66

SEZNAM TABULEK

Tabulka 1: URI a odkazy na doporučení.....	14
Tabulka 2: Třídy a odpovědi.....	22
Tabulka 3: Základní příkazy v Asterisk konzoli.....	54

SEZNAM PŘÍLOH

Příloha P 1: Konfigurační soubor `openser.cfg`.

Příloha P 2: `SIP.conf`

Příloha P 3: `extensions.conf`

PŘÍLOHA P 1: KONFIGURAČNÍ SOUBOR OPENSER.CFG.

```
##### Global Parameters #####

debug=4
log_stderr=no
log_facility=LOG_LOCAL0

fork=yes
children=4

listen=udp:62.44.19.70:5060
disable_tcp=yes
mhomed=1

##### Modules Section #####

# Nastavení cesty k modulům
mpath="/usr/lib/openser/modules/"

/* uncomment next line for MySQL DB support */
loadmodule "mysql.so"
loadmodule "sl.so"
loadmodule "tm.so"
loadmodule "rr.so"
loadmodule "maxfwd.so"
loadmodule "usrloc.so"
loadmodule "registrar.so"
loadmodule "textops.so"
loadmodule "mi_fifo.so"
loadmodule "uri_db.so"
loadmodule "uri.so"
loadmodule "avpops.so"
loadmodule "domain.so" #pridano1
loadmodule "nathelper.so" #pridano1
loadmodule "mediaproxy.so" #pridano1
loadmodule "xlog.so"
loadmodule "acc.so"
loadmodule "auth.so"
loadmodule "auth_db.so"

# ----- setting module-specific parameters -----

# ----- mi_fifo params -----
modparam("mi_fifo", "fifo_name", "/tmp/openser_fifo")

modparam("rr", "enable_full_lr", 1)
modparam("rr", "append_fromtag", 0)
modparam("avpops", "avp_table", "usr_preferences") #PRIDANO_1

# ----- rr params -----
modparam("usrloc", "nat_bflag", 6) #PRIDANO_podpora NAT
modparam("nathelper", "sipping_bflag",7) #PIRDANO_podpora NAT
modparam("nathelper", "rtpproxy_sock",
"unix:/var/run/rtpproxy/rtpproxy.sock")

# ----- uri_db params -----
modparam("uri_db", "use_uri_table", 0)
```

```

modparam("uri_db", "db_url", "")

# ----- acc params -----
modparam("acc", "radius_config", "")
modparam("acc", "early_media", 1)
modparam("acc", "report_ack", 1)
modparam("acc", "report_cancels", 1)
modparam("acc", "detect_direction", 0)
modparam("acc", "failed_transaction_flag", 3)
modparam("acc", "log_flag", 1)
modparam("acc", "log_missed_flag", 2)
modparam("acc", "db_flag", 1)
modparam("acc", "db_missed_flag", 2)

# ----- auth_db params -----
modparam("auth_db", "calculate_ha1", yes)
modparam("auth_db", "password_column", "password")

modparam("nathelper", "rtpproxy_sock", "udp:127.0.0.1:22222")
modparam("nathelper", "natping_interval", 30)
modparam("nathelper", "ping_nated_only", 1)
modparam("nathelper", "sipping_from", "openser:openser@localhost")

# ----- Tomikova_domena -----
modparam("usrloc", "db_url", "mysql://openser:HESLO@localhost/openser")
modparam("usrloc", "db_mode", 2)

##### Routing Logic #####

# main request routing logic
route{

    if (uri==myself) {
        xlog("L_ERR", "SIP request [$rm] to my domain [$rd]\n");
    } else {
        xlog("L_ERR", "SIP request [$rm] to foreign domain [$rd]\n");
    };

    if (!mf_process_maxfwd_header("10")) {
        sl_send_reply("483", "Too Many Hops");
        exit;
    }

    if (has_totag()) {
        # sequential request withing a dialog should
        # take the path determined by record-routing
        if (loose_route()) {
            if (is_method("BYE")) {
                setflag(1); # do accounting ...
                setflag(3); # ... even if the transaction fails
            }
            route(1);
        } else {
            /* uncomment the following lines if you want to enable
presence */
            ##if (is_method("SUBSCRIBE") && $rd ==
"your.server.ip.address") {
            ##     # in-dialog subscribe requests
            ##     route(2);

```

```

        ##      exit;
        ##}
        if ( is_method("ACK") ) {
            if ( t_check_trans() ) {
                # non loose-route, but stateful ACK; must
                # be an ACK after a 487 or e.g. 404 from upstream server
                t_relay();
                exit;
            } else {
                # ACK without matching transaction ...
                ignore and discard.\n");
                exit;
            }
        }
        sl_send_reply("404","Not here");
    }
    exit;
}

#initial requests

# CANCEL processing
if (is_method("CANCEL"))
{
    if (t_check_trans())
        t_relay();
    exit;
}

t_check_trans();

# record routing
if (!is_method("REGISTER|MESSAGE"))
    record_route();

# account only INVITES
if (is_method("INVITE")) {
    setflag(1); # do accounting
}
if (!uri==myself)
# requests for my domain

if (is_method("PUBLISH"))
{
    sl_send_reply("503", "Service Unavailable");
    exit;
}
if (is_method("REGISTER"))
{
    if (!save("location"))
        sl_reply_error();

    exit;
}

if ($rU==NULL) {
    # request with no Username in RURI
    sl_send_reply("484","Address Incomplete");
    exit;
}

```

```

# apply DB based aliases (uncomment to enable)
##alias_db_lookup("dbaliases");

if (!lookup("location")) {
    switch ($retcode) {
        case -1:
        case -3:
            t_newtran();
            t_reply("404", "Not Found");
            exit;
        case -2:
            sl_send_reply("405", "Method Not Allowed");
            exit;
    }
}

# when routing via usrloc, log the missed calls also
setflag(2);

route(1);

if (uri==myself) {
    if (method=="REGISTER") {
        # Uncomment this if you want to use digest
        authentication
        {
            #if (!www_authorize("openser.org", "subscriber"))
            {
                # www_challenge("openser.org", "0");
                # exit;
                #};

                save("location");
                exit;
            };
            lookup("aliases");
            if (!uri==myself) {
                append_hf("P-hint: outbound alias\r\n");
                route(1);
            };

            if (uri=~"sip:0.*@.*") {
                append_rpid_hf(); # for submitting external number from
                customers from subscriber table/rpid
                route(3); # Asterisk route
                exit;
            };

            # native SIP destinations are handled using our USRLOC DB
            if (!lookup("location")) {
                sl_send_reply("404", "Not Found");
                exit;
            };
            append_hf("P-hint: usrloc applied\r\n");
        };
        route(1);
    }
}
route[1] {
    # for INVITEs enable some additional helper routes
    if (is_method("INVITE")) {

```

```

        t_on_branch("2");
        t_on_reply("2");
        t_on_failure("1");
    }

    if (!t_relay()) {
        sl_reply_error();
    };
    exit;
}
# Presence route
/* uncomment the whole following route for enabling presence
   NOTE: do not forget to enable the call of this route from the main
   route */
##route[2]
##{
##    if (!t_newtran())
##    {
##        sl_reply_error();
##        exit;
##    };
##
##    if(is_method("PUBLISH"))
##    {
##        handle_publish();
##        t_release();
##    }
##    else
##    if( is_method("SUBSCRIBE"))
##    {
##        handle_subscribe();
##        t_release();
##    }
##
##    exit;
##}

#branch_route[2] {
#    xlog("new branch at $ru\n");
#}

onreply_route[2] {
    xlog("incoming reply\n");
}

/*failure_route[1] {
    if (t_was_cancelled()) {
        exit;
    }

    # uncomment the following lines if you want to block client
    # redirect based on 3xx replies.
    ##if (t_check_status("3[0-9][0-9]")) {
    ##t_reply("404","Not found");
    ##    exit;
    ##}

    # uncomment the following lines if you want to redirect the failed
    # calls to a different new destination

```

```

    ##if (t_check_status("486|408")) {
    ##    sethostport("192.168.2.100:5060");
    ##    append_branch();
    ##    # do not set the missed call flag again
    ##    t_relay();
    ##}
}*/
route[3] {

# -----
# ASTERISK Handler
# -----
xlog("L_INFO","Step 500 Entering ==== Route 3 (Asterisk) =====");
rewritehostport("62.44.19.70:5065"); # INSERT YOUR Asterisk GATEWAY IP
ADDRESS and port here
route(1); # now use standort route with t_relay
}

route[4] {
# -----
# NAT Traversal Section
# -----

if (isflagset(6)) {

force_rport();

fix_nated_contact();

force_rtp_proxy();

};

}

```

PŘÍLOHA P 2: SIP.CONF

```
[general]
bindport=5060
bindaddr=0.0.0.0
;domain=voip-iglo.cz
disallow=all
allow=ulaw
allow=alaw
allow=gsm

register => 511119595:heslo@212.71.146.175:5060/511119595

[517350051]
type=friend
fromuser=02_pstn
username=02_pstn
secret=heslo
host=192.168.1.2
port=5061
dtmfmode=rfc2833
insecure=port,invite
context=main

[spa]
type=friend
host=192.168.1.2
context=main
username=line1
userid=line1 <123>
secret=heslo
dtmfmode=rfc2833
disallow=all
allow=ulaw

[802_voip]
type=friend
host=hlas.802.cz
username=511119595
fromuser=511119595
secret=heslo
fromdomain=hlas.802.cz
dtmfmode=rfc2833
allow=alaw
allow=ulaw
permit=212.71.146.175
insecure=invite
context=main

[222]
type=friend
username=222
userid=test2 <222>
host=dynamic
secret=heslo
context=main

[444]
type=friend
username=444
```



```
userid=tomikz <444>  
nat=yes  
host=dynamic  
secret=heslo  
context=inside  
  
[111_echo]  
type=friend  
username=222  
userid=echotest <111>  
host=dynamic  
secret=heslo
```

PŘÍLOHA P 3: EXTENSIONS.CONF

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

[globals]
sounds_exten = /usr/share/asterisk/sounds/
sounds_phonetic = /usr/share/asterisk/sounds/phonetic/
sounds_letters = /usr/share/asterisk/sounds/letters/
sounds_silence = /usr/share/asterisk/sounds/silence/

[macro-enum]
; ARG1 je hledane cislo, ARG2 je schema (sip, iax2)
exten => s,1,Set(sipcount=${ENUMLOOKUP(+${ARG1},${ARG2},c)}|counter=0)
exten => s,n,GotoIf("${counter}"<"${sipcount}")?start:end)
exten => s,n(start),Set(counter=${counter}+1))
exten => s,n,Dial(SIP/${ENUMLOOKUP(+${ARG1},${ARG2},${counter}))
exten => s,n,GotoIf("${counter}"<"${sipcount}")?start:end)
exten => s,n(end),NoOp("Zadne validni ${ARG2} zaznamy v e164.arpa pro ${ARG1}")

[main]
exten => _111,1,Dial(SIP/111,10)
exten => _111,n,NoOP(doleze to sem?)
exten => _111,n,goto(echo,s,1)
exten => _XXX,1,Dial(SIP/${EXTEN})
exten => 511119595,1,Dial(SIP/222,20)
exten => 511119595,n,Playback(vm-nobodyavail)
exten => 123,1,NoOP(${CALLERID})
exten => 123,n,Answer()
exten => 123,n,Ringing()
exten => 123,n,Wait(8)
exten => 123,n,Dial(SIP/222)
exten => 123,n,Hangup()

;volani na cisla
exten => _1XXXXXXXXXX,1,NoOp
exten => _1XXXXXXXXXX,n,Dial(SIP/${EXTEN:1}@517350051,30,r)
exten => _1XXXXXXXXXX,n,Hangup

; tisnove linky volame primo
exten => _112,1,Dial(${OUTBOUNDTRUNK}/${EXTEN})
exten => _158,1,Dial(${OUTBOUNDTRUNK}/${EXTEN})
exten => _156,1,Dial(${OUTBOUNDTRUNK}/${EXTEN})
exten => _150,1,Dial(${OUTBOUNDTRUNK}/${EXTEN})
exten => _155,1,Dial(${OUTBOUNDTRUNK}/${EXTEN})

; ENUM pro mezinarodni cisla
; je zapotrebi odstranit dve nuly na zacatku
exten => _00X.,1,Macro(enum,${EXTEN:2},sip)
exten => _00X.,n,NoOp("MEZINARODNI FORMAT - OREZ 00")
exten => _00X.,n,Macro(enum,${EXTEN:2},iax2)
exten => _00X.,n,Dial(${OUTBOUNDTRUNK}/${EXTEN})

; ENUM pro ceska cisla
exten => _X.,1,Macro(enum,${EXTEN:0},sip)
exten => _X.,n,NoOp("CESKY FORMAT - NIC NEOREZE")
exten => _X.,n,Macro(enum,${EXTEN:0},iax2)
; ostatni - volame pres PSTN
```

```
exten => _X.,n,Dial(${OUTBOUNDTRUNK}/${EXTEN})
exten => _X.,n,Dial(SIP/${EXTEN:0}@802_voip,,r)

[echo]
exten => s,1,Playback(demo-echotest)
exten => s,n,Echo()
exten => s,n,Playback(demo-echodone)
exten => s,n,Goto(s,1)

[inside]
;exten => _XXX,1,Dial(SIP/${EXTEN}@62.44.19.70,10) #Vyzvaneni po dobu
10vterin, pote bude presmerovano

;waiting_unavailable
exten => _XXX,1,Dial(SIP/${EXTEN},10)
exten => _XXX,n,Goto(s-${DIALSTATUS},1) #Skok na NOANSWER odpovedi
(uzivatel neni dostupny)
exten => s-NOANSWER,1,Playback(vm-nobodyavail)
exten => s-NOANSWER,2,Wait(1)
exten => s-NOANSWER,3,Playback(vm-toreply)
exten => s-NOANSWER,4,Hangup()
```