

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV INFORMAČNÍCH SYSTÉMŮ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

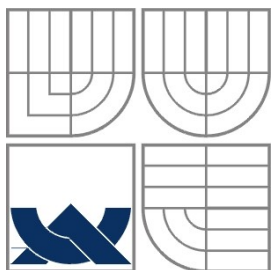
## **VYUŽITÍ SIP SERVERU NA FIT PRO IP TELEFONII**

**DIPLOMOVÁ PRÁCE**  
MASTER'S THESIS

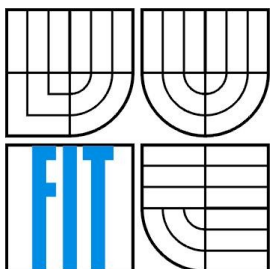
**AUTOR PRÁCE**  
AUTHOR

Bc. Lukáš Hýbner

BRNO 2008



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV INFORMAČNÍCH SYSTÉMŮ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

## **VYUŽITÍ SIP SERVERU NA FIT PRO IP TELEFONII**

DEPLOYMENT OF SIP SERVER AT THE FIT FOR IP TELEPHONY

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

Bc. Lukáš Hýbner

**VEDOUCÍ PRÁCE**

SUPERVISOR

Ing. Petr Matoušek, Ph.D.

BRNO 2008

# Využití SIP serveru na FIT pro IP telefonii

## Prohlášení

Prohlašuji, že jsem tento semestrální projekt vypracoval samostatně pod vedením

Ing. Petra Matouška, Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Lukáš Hýbner  
30.7.2008

© Lukáš Hýbner, 2008

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

## **Abstrakt**

Práce se zabývá možností připojení serveru SIP do telefonní sítě FIT. Účelem je, aby mohli volat zaměstnanci na univerzitu, pokud jsou mimo fakultu. Pro řešení využijeme SIP server Asterisk, který bude sloužit jako autorizační server pro uživatele. Dále server bude zajišťovat převod čísel na adresu SIP s využitím ENUM. V praktické části ověříme funkčnost, toho systému.

## **Klíčová slova**

VoIP, SIP, ENUM, Asterisk, IP telefonie

## **Abstract**

Master's thesis is engaged on possibilities connect SIP server to telephone network on FIT. The main reason is that employees can call to university, when they are out of faculty. For resolution we will use SIP server Asterisk, which will be serving as authorization server for users. Next Asterisk will ensure transmission numbers to SIP address with ENUM. In the practical part we will verify the functionality.

## **Keywords**

VoIP, SIP, ENUM, Asterisk, IP telephony

# Obsah

Obsah.....	1
Seznam tabulek a obrázků.....	3
1 Úvod.....	4
2 SIP protokol.....	6
2.1 Adresování.....	7
2.2 Druhy zařízení v SIP síti.....	7
2.2.1 UA zařízení.....	8
2.2.2 SIP servery.....	8
2.3 Formát komunikace.....	12
2.3.1 Žádost.....	12
2.3.2 Odpověď.....	13
2.3.3 Formát zpráv.....	14
2.3.4 Transakce.....	17
2.3.5 Dialog.....	18
2.3.6 Registrace a autentizace uživatelů.....	19
2.4 Směrování.....	20
3 Asterisk.....	22
3.1 Služby ústředny.....	22
3.2 Vnitřní architektura.....	23
3.3 Číslovací plán (Dial plan).....	26
3.4 Architektura SIP v Asterisku.....	27
3.5 SIP a NAT v Asterisku.....	28
3.5.1 Server za NATem.....	29
3.5.2 Klient za NATem.....	30
4 Propojení SIP serveru s dosavadní IP telefoníí na FIT.....	33
4.1 Adresování v telefonii a standard ENUM.....	33
4.2 ENUM v ČR.....	34
4.3 Mapování telefonního čísla pomocí ENUM.....	35
4.4 DNS záznam typu NAPTR.....	35
4.4.1 Struktura NAPTR záznamů.....	35
4.4.2 Algoritmus výběru záznamu NAPTR.....	37

4.4.3 NAPTR záznamy s ENUM.....	38
4.5 DNS SRV záznamy se SIP.....	39
4.6 Nastavení Asterisku na překlad ENUM.....	40
5 Testování.....	42
5.1 Testování ENUM.....	42
5.1.1 Ukázka SIP komunikace se zamítnutím.....	43
5.1.2 Ukázka SIP komunikace s telefonním číslem.....	44
6 Akademická síť CESNET.....	45
6.1 Struktura VoIP v síti CESNET.....	45
6.2 Podpora ENUM.....	46
6.3 Možnosti propojení.....	46
7 Závěr.....	47
Literatura.....	48
Seznam příloh.....	49

# Seznam tabulek a obrázků

## Seznam tabulek

Tabulka 1: Souhrn hlasových kodeků.....	25
Tabulka 2: Konfigurační soubory serveru Asterisk.....	25
Tabulka 3: Konfigurace NATu v souboru sip.conf.....	29
Tabulka 4: Hodnoty pole Flags v záznamu NAPTR.....	36
Tabulka 5: Služby systému ENUM.....	38

## Seznam obrázků

Obrázek 1: Komunikace SIP klientů.....	8
Obrázek 2: Volání přes proxy server.....	10
Obrázek 3: Registrace klienta SIP.....	11
Obrázek 4: Přesměrování klienta SIP.....	11
Obrázek 5: Ukázka SIP komunikace.....	14
Obrázek 6: Transakce SIP.....	18
Obrázek 7: Registrace klienta na SIP server.....	20
Obrázek 8: Architektura serveru Asterisk.....	24
Obrázek 9: Struktura paketu.....	28
Obrázek 10: Připojení serveru Asterisk za NATem.....	30
Obrázek 11: Volba canreinvite s hodnotou „no“.....	32
Obrázek 12: Volba canreinvite s hodnotou „yes“.....	32
Obrázek 13: Vytočení čísla s ENUM překladem.....	43
Obrázek 14: Struktura VoIP sítě CESNET.....	45

# 1 Úvod

IP telefonie si za dobu své existence vydobývá místo na internetu. V dnešní době informačních technologií a stále vyšší rychlosti internetu, je dobré využívat internet nejen jako zdroj informací, ale také jako možnost hlasové komunikace. Pokud se podíváme na dnešní ceny paušálů a ceny volání dominantního operátora pevných linek, tak jsou ty ceny velmi vysoké, v porovnání s IP telefonii. Dále jsou zde neporovnatelné ceny za spojení hovorů do zahraničí. Pokud tedy někdo ze zaměstnanců FITu má doma internet (nebo v hotelu na služebních cestách), tak je velmi jednoduché použít IP telefonii na volání do VUT sítě.

Cílem této práce je nakonfigurovat SIP server Asterisk, který bude komunikovat s VoIP sítí FIT. A dále, aby se na tento server mohli přihlašovat zaměstnanci FIT, kteří potom budou moci telefonovat na VUT zdarma. V současnosti když volá někdo na VUT může si vybrat, přes jakého operátora bude volat. Stačí znát pouze linku (poslední čtyři číslice) a předčíslí je dané podle operátora. Např. pro Telefoniku O2 je to 54114, pro T-mobile je to 60407, atd. Všechny tyto možnosti stojí peníze. Praktická část této práce se bude zabývat konfigurací serveru Asterisk, kde nastavím překlad telefonních čísel na adresy SIP pomocí systému ENUM.

V další kapitole se zabývám SIP protokolem. Je to moderní a nejpoužívanější VoIP protokol, proto se jím zabývám v samostatné kapitole. Procházím jeho vývojem a popisuji adresování, druhy zařízení, formát komunikace a směrování.

Ve třetí kapitole detailně rozebírám konfiguraci SIP protokolu v serveru Asterisk. Rozebírám služby serveru, instalace a další. Dále se v kapitole zabývám, jak umožnit a nakonfigurovat server Asterisk pro průchod SIP protokolu přes NAT. Další zajímavá část je, jak se Asterisk konfiguruje v roli klienta SIP serveru.

Ve čtvrté kapitole shrnuji možnosti připojení do sítě FIT. Popisuji zde systém ENUM, jeho možnosti a stav v České republice. Dále detailně popisuji DNS záznamy NAPTR, které slouží pro systém ENUM. Další podkapitola se zabývá DNS záznamy SRV, které slouží pro informace o dostupných službách na dané doméně. Dále rozebírám konfiguraci ENUM v serveru Asterisk.



V páté kapitole jsem nastínil, jaké jsem měl praktické problémy při testování překladač telefonních čísel pomocí systému ENUM na adresu SIP v serveru Asterisk.

Šestá kapitola je zaměřena na akademickou síť CESNET. Je zde uvedena základní struktura VoIP sítě CESNETu. Dále je zde popsáno jak síť CESNET podporuje standard ENUM.

## 2 SIP protokol

Tato kapitola má ukázat základní vlastnosti SIP protokolu a jeho princip. SIP protokol je signalizační protokol, který se využívá pro signalizaci hovorů. Tento protokol se využívá na FIT, proto jej zde budu popisovat. Existují také jiné signalizační protokoly např. H323, který byl velmi oblíbený v době, kdy ještě neexistoval SIP.

Tento protokol se začal vyvíjet v roce 1996, později byl přijat internetovou komunitou. V současnosti se využívá jeho druhá verze z roku 2002 dle RFC 3261 [1]. Protokol se využívá pro signalizaci hovorů. Je to jednoduchý aplikační protokol pro inicializaci, modifikaci a ukončování interaktivních relací. Byl navržen tak, aby byl jednoduše rozšiřitelný a flexibilní. Pro komunikaci jsou také důležité další protokoly, je to zejména RTP a SDP. RTP se používá pro přenos hlasových paketů v reálném čase. SDP se využívá pro popis vlastností klientů. Popis vlastností je využíván pro parametry spojení, jako jsou kodeky, transportní protokol, atd..

SIP je end-to-end orientovaný protokol. Znamená to, že logika protokolu je na koncovém zařízení a zařízení zná i aktuální stav komunikace, což je výhodné pro zvýšení odolnosti oproti chybám. Nevýhodou tohoto řešení je vyšší režie při zasílání zpráv mezi koncovými zařízeními. SIP byl navržen tak, aby zajistil stejnou funkcionalitu jako PSTN (klasické telefonní síť s přepínáním okruhů), ale zároveň díky návrhu end-to-end je otevřena možnost k implementaci nových služeb, které nelze nasadit v sítích PSTN.

Komunikace v SIPu je založena na zprávách typu požadavek a odpověď. SIP vychází z HTTP protokolu. Oba protokoly mají obdobný formát hlaviček zpráv. HTTP lze taktéž považovat za signalizační protokol, protože jej klienti využívají, aby požádali HTTP server o nějaký dokument. Díky formátu protokolu je SIP oblíbený mezi vývojáři aplikací i mezi správci VoIP sítí, což vyplývá z jeho textové povahy a jednoduchosti.

SIP protokol umožňuje následující činnosti:

- lokalizaci uživatele
- navázání spojení
- zjištění dostupnosti uživatele

- možnosti klienta (např.: podporované příkazy, podporovaný hlasový kodek, apod.)
- řízení spojení – ukončení spojení

Naproti tomu neumí zajistit požadovanou QoS, protože neumí upřednostňovat nějaký provoz ani rezervovat potřebné síťové prostředky. Spolupracuje však s protokoly, které se o zajištění QoS umějí postarat.

## 2.1 Adresování

Adresa SIP je definována v RFC 3261 [1] a skládá se ze dvou částí, uživatele a názvu domény. Tato adresa reprezentuje uživatele z určité domény, kterou obsluhuje SIP server. K adresaci se používá SIP URI. Formát URI je následující:

sip:uživatel@host:port;parametry (např: sip:541141108@vutbr.cz)

Jak je patrné adresa SIP se velmi podobá emailové adrese a je dobré používat stejnou adresu pro email a SIP. Část uživatel reprezentuje konkrétního uživatele z domény, která je identifikována částí host. Část host je buď doménové jméno nebo IP adresa. Dále následuje číslo portu, když není uvedeno předpokládá se standardní 5060. Parametry mohou být použity pro další informace.

Díky takovému způsobu adresace mohou uživatelé kdekoliv na internetu komunikovat s kýmkoliv. IP zařízení s podporou SIP mohou komunikovat přímo, pokud znají URI druhé strany.

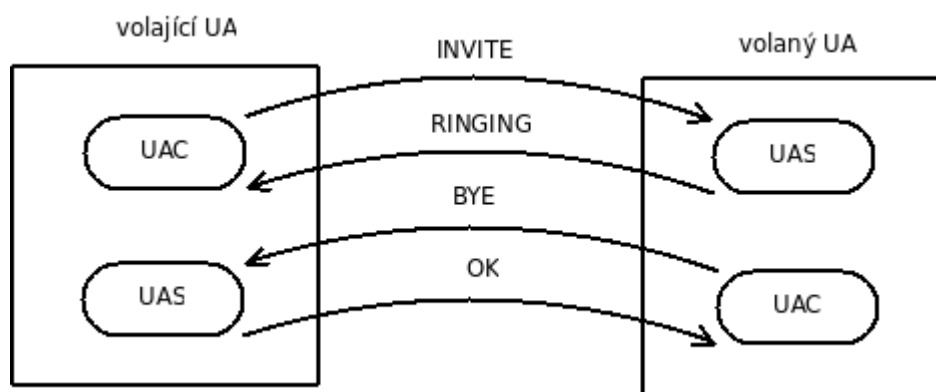
## 2.2 Druhy zařízení v SIP síti

Již bylo řečeno, že komunikace probíhá mezi dvěma koncovými SIP zařízeními, ale je možné, aby SIP síť obsahovala více druhů zařízení. Zařízení jsou dle RFC 3261 [1] následující:

- UA (User Agent): je to koncové zařízení
- různé typy serverů (registrar, proxy, redirect)

## 2.2.1 UA zařízení

UA je koncové zařízení, typicky HW SIP telefon nebo softwarová aplikace. Je složena ze dvou částí UAS (User Agent Server) a UAC (User Agent Client). Tyto dvě části jsou pouze logické entity a každý UA je obsahuje. UAC je část, která vyšle požadavek a čeká na odpověď. UAS je část, která vyčkává na požadavek a jakmile ho přijme vyšle na něj odpověď. Na následujícím obrázku je vidět příklad. Volající UA je ve funkci UAC, když odesílá požadavek na sestavení hovoru (INVITE) a čeká na odpověď (RINGING). Zatímco volaný UA je ve funkci UAS, když přijímá požadavek na sestavení hovoru a následně odešle odpověď. Dále je vidět, že když volaný ukončuje spojení (BYE), tak je ve funkci UAC. Volající zprávu potvrzuje (OK), a tak je ve funkci UAS.



Obrázek 1: Komunikace SIP klientů

## 2.2.2 SIP servery

Pomocí SIP protokolu můžeme vytvářet síťovou infrastrukturu různých typů SIP serverů. Typy serverů jsou logické části, které lze provozovat společně na jednom fyzickém stroji nebo na více strojích. Většinou se používá jeden fyzický stroj kvůli efektivnosti. Jednotlivé typy jsou následující:

- proxy: Server zastupuje UA při předávání požadavků na další server nebo další UA
- registrar: Zajišťuje registraci momentálního umístění klientů.
- redirect: Informují klienta o skoku v síti, kam se má zpráva poslat a klient nebo proxy následně kontaktuje server sám.

## Proxy server

Proxy servery jsou důležitým prvkem v SIP infrastruktuře. Jeho typická role spočívá ve směrování žádostí o spojení podle toho, kde je umístěno adresované UA. Při této žádosti bude prohledávat další proxy servery, dokud nenalezne nějaký, který zná umístění adresovaného UA. Typy SIP proxy serverů:

- stateless (bezstavový)
- stateful (stavový)

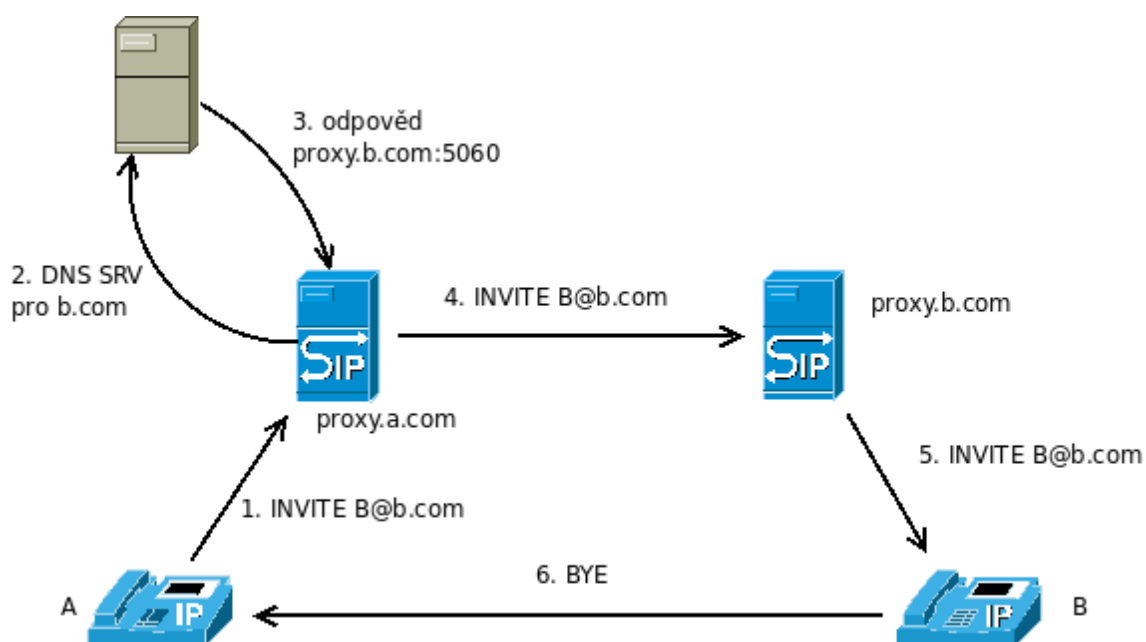
Typická role bezstavového proxy serveru je v přeposílání zpráv, bez ohledu na jejich význam. Díky tomu je tento proxy server nenáročný a může obsloužit mnoho klientů. Nevýhodou je neumí kontrolovat smysluplnost zpráv, a tak mohou nastat nekorektní stavy, ty jsou ale ošetřeny UA. Nemohou také zachytit opakování zpráv nebo větvení při směrování. Jejich využití je především pro snížení zátěže nebo jednoduché směrování.

Stavový proxy server si na začátku komunikace uloží informaci o stavu a udržuje ji během transakce. Některé transakce mohou trvat déle, např. vyhledání uživatele může trvat déle. Kvůli tomu je výkon limitován. Naproti tomu stavový proxy server má některé zajímavé vlastnosti jako je větvení hovorů při směrování. Například může nejdříve vytočit adresu do zaměstnání a v případě nevyzvednutí žádost o spojení přesměrovat na adresu domů. Dále může zachytit opakování zpráv od uživatele. Většina proxy serverů je stavová, často zajišťuje účtování a má zabudovanou podporu NATu. Stavový proxy server poznáme, budeme-li zachycovat signalizaci. Pokud nejdříve proxy server na žádost odpoví a až následně ji přepoše, tak se jedná o stavový proxy server. Pokud bude proxy server pouze přeposílat zprávy, tak se jedná o bezstavový proxy server.

Obvyklá situace je, že firma vlastní svůj proxy server, který obsluhuje všechny UA v dané firmě. Většinou proxy server obsluhuje jednu doménu, může ale obsluhovat i více domén. V takovém případě obsahuje pole *realm* ve zprávě SIP doménu a proxy server umí určit do jaké domény patří uživatel. Na následujícím obrázku vidíte situaci dvou firem A a B. Každá firma má svůj proxy server, u firmy A je to proxy.a.com a u firmy B je to proxy.b.com. Uživatel A (firma A) chce zavolat uživateli B (z firmy B). Jednotlivé kroky jsou následující:

1. Uživatel A použije adresu sip:b@b.com a pošle žádost o spojení. Protože UA uživatele A neví, kam má poslat danou žádost, tak ji pošle svůj výchozí proxy server.

2. Proxy server když zjistí, že uživatel sip:b@b.com je z jiné domény, tak pošle DNS SRV dotaz na doménu b.com, pokud není záznam zadán staticky přímo v proxy serveru.
3. Proxy server dostane odpověď od DNS serveru.
4. Následně pošle žádost o spojení na server proxy.b.com.
5. Proxy server firmy B dostane žádost a pošle ji na UA uživatele B.
6. Uživatel B ukončí hovor. Tuto zprávu může poslat přímo uživateli A, protože zná je IP adresu, kterou dostal v žádosti o spojení v poli *Contact*. Podrobněji bude probráno v kapitole o směrování.

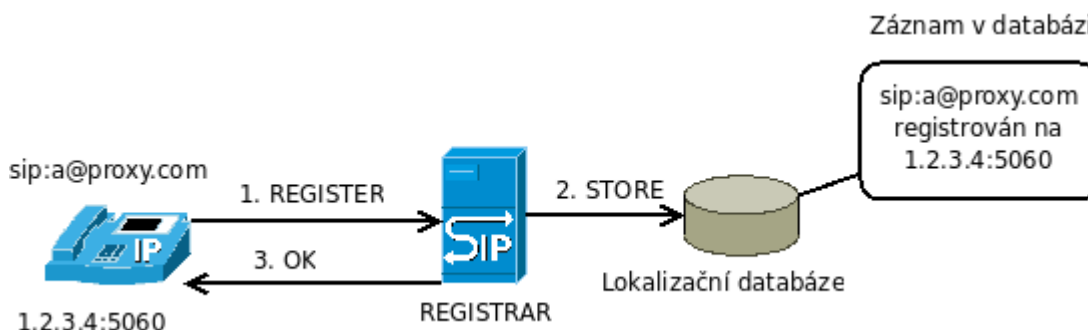


Obrázek 2: Volání přes proxy server

## Registrar server

Další důležitý prvek v infrastruktuře SIP je registrar server. Abychom mohli uživatele lokalizovat, musí se někde uživatel registrovat na registrar server. Role registrar serveru tedy spočívá v tom, že uchovává informace v lokalizační databázi o tom, kdo je přihlášen a na jaké IP adrese se nachází. Dále uchovává informace o času přihlášení, protože pokud UA po nějaké době neobnoví registraci, tak registrace vyprší. Vypršení registrace znamená, že se na daného klienta nelze dovolat, protože registrar server označí informaci o jeho poloze za neaktuální.

Na následujícím obrázku vidíme, jak se klient registruje na registrar server zprávou REGISTER. Následně si registrar server uloží čas registrace, IP adresu 1.2.3.4 a port 5060 k adrese SIP sip:a@proxy.com. Nakonec server potvrdí registraci zasláním zprávy OK.

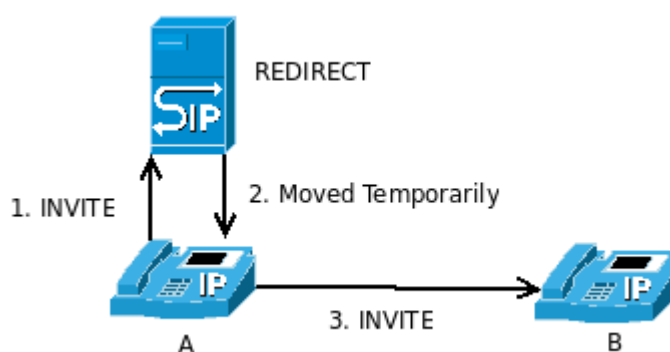


Obrázek 3: Registrace klienta SIP

## Redirect server

Redirect server přijímá požadavky na lokalizaci uživatele. Pak se podívá do lokalizační databáze registrar serveru a vytvoří seznam lokalizací uživatele. Následně posílá odpověď obsahující přesměrování se seznamem lokalizací uživatele. Odesílatel potom pomocí tohoto seznamu odesílá další požadavky přímo podle seznamu.

Na následujícím obrázku vidíme typickou situaci při přesměrování. Uživatel A pošle žádost o spojení s uživatelem B na redirect server a ten mu vzápětí pošle seznam aktuálního umístění uživatele B s příkazem přesměrování. Následně potom je poslána nová žádost o spojení přímo na uživatel B. Pokud redirect server pošle seznam obsahující více záznamů, tak může klient poslat žádosti o spojení postupně nebo najednou na všechny umístění a následně vybrat první, který odpoví. Ostatním pak pošle zprávu Cancel.



Obrázek 4: Přesměrování klienta SIP

## 2.3 Formát komunikace

Při SIP komunikaci se používají zprávy, které mají jednoduchý textový formát, podobně jako HTTP, což je výhodné při testování a rozšiřování protokolu. SIP není určen pro přenos velkého objemu dat, místo toho přenáší pouze malý objem dat, potřebný pro navázání interaktivních relací. Je také schopen přenášet krátké textové zprávy. Zprávy SIP se mohou přenášet pomocí TCP nebo UDP a většinou se používá UDP. Každá zpráva obsahuje hlavičku a tělo zprávy. Zprávy se dělí na dva typy:

- žádost
- odpověď

### Ukázka zprávy:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.22.1.103:5060;branch=z9hG4bK1945437743;rport=5060
From: "200" <sip:200@172.22.1.1>;tag=852125087
To: "200" <sip:200@172.22.1.1>;tag=as4aee0161
Call-ID: 1765981028@172.22.1.103
CSeq: 5641 REGISTER
User-Agent: Asterisk PBX
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY
Expires: 60
Contact: <sip:200@172.22.1.103:5060>;expires=60
Content-Length: 0
```

V žádostech se používají příkazy, které slouží především na sestavení nebo ukončení spojení, případně mohou oznamovat nějaký požadavek. Odpovědi se používají jako potvrzení, že žádost byla doručena a zpracovává se, obsahují také návratový kód zpracování.

### 2.3.1 Žádost

Zprávy SIP typu žádost obsahují příkazy. Základní příkazy jsou:

- REGISTER – Registruje klienta na SIP server. Zpráva s tímto příkazem obsahuje IP adresu a port UA, která slouží pro nastavení aktuální polohy uživatele na serveru. Registrace je časově omezena a musí se po nějakém čase obnovovat.
- INVITE – Je to příkaz k sestavení spojení nebo změnu parametrů již probíhajícího spojení.



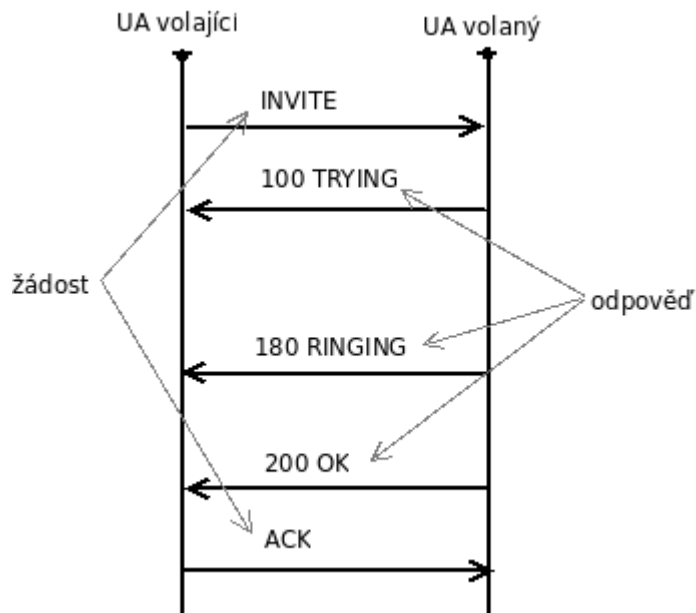
- ACK – Zpráva obsahující tento příkaz znamená, potvrzení přijetí odpovědi na žádost INVITE. Sestavení spojení používá synchronizace 3-way handshake, kdy na příkaz INVITE, odpoví adresát zasláním odpovědi OK a to opakuje, dokud nedostane žádost ACK. Na tuto žádost se již neodpovídá.
- CANCEL – Zrušení sestavování spojení hovoru. Používá ji volající, pokud volaný ještě nepotvrdil příkaz INVITE.
- BYE – Ukončení probíhajícího spojení.
- OPTIONS – Příkaz k zaslání vlastností serveru nebo UA.

## 2.3.2 Odpověď

Odpověď odesílá proxy server nebo UA na každou zprávu typu žádost. Výjimku tvoří žádost s příkazem ACK. V odpovědích se používají návratové kódy, které vycházejí z HTTP. Využívá se zde tzv. stovkových kódů. Číselné kódy mají textovou reprezentaci, která se uvádí hned za nimi v textové zprávě. Rozdělení kódů je následující:

- 1xx – Tyto kódy obsahují odpovědi odesílající se na žádost, které byly přijaty, ale výsledek zpracování ještě není znám. Díky této odpovědi musí odesílatel přestat opakovat zaslání dané žádosti. Při zpracování příkazu INVITE proxy server obvykle odesílá zprávu se kódem 100 (Trying) a UA odesílá zprávu se kódem 180 (Ringing), která značí vyzvánění.
- 2xx – Zprávy s tímto kódem značí úspěšné zpracování konkrétní žádosti. Tuto odpověď dostává odesílatel jako poslední na zpracování své žádosti. Na zpracování příkazu INVITE posílá UA odpověď s kódem 200 (OK), jestliže přijme hovor.
- 3xx – Odpovědi s tímto kódem jsou používány pro přesměrování. Tyto odpovědi informují odesílatele o nové poloze uživatele nebo jiné dostupné službě, která může být použita. Když proxy server není schopen zpracovat žádost, tak posílá odesílateli odpověď s kódem o přesměrování a vloží tam i aktuální umístění. Může to být proxy server nebo aktuální umístění z lokalizační databáze vytvořené při registraci. Odpověď s tímto kódem je poslední při zpracování žádosti. Volající pak kontaktuje uživatele na novém umístění tím, že vyše novou žádost.
- 4xx – Zprávy s tímto kódem znamenají, že požadavek je chybný a nelze jej zpracovat, nebo problém na straně klienta. Například klient není autorizován nebo je obsazeno.
- 5xx – Odpověď s tímto kódem znamená, že požadavek nelze zpracovat, nastala chyba na proxy serveru. Klient může zaslat požadavek znovu později.

- 6xx – Pokud nelze žádost splnit na žádném serveru, tak odpověď obsahuje tento kód. Odpověď s kódem 603 (Decline response) zasílá server, pokud UA odmítá žádost o spojení.



Obrázek 5: Ukázka SIP komunikace

### 2.3.3 Formát zpráv

Zprávy SIP jsou rozděleny na hlavičku a tělo. Hlavička zprávy SIP typu žádost má na prvním řádku příkaz s parametrem Request-URI a na konci řádku je označení protokolu SIP/2. Hlavička zprávy SIP typu odpověď obsahuje na prvním řádku označení protokolu SIP/2 a následuje návratový kód s názvem.

#### Ukázka hlavičky zprávy:

```

INVITE sip:200@172.22.1.103:5060 SIP/2.0
Via: SIP/2.0/UDP 172.22.1.65:5060;branch=z9hG4bK2918f68b;rport
From: "100" <sip:100@172.22.1.65>;tag=as44d4d98e
To: <sip:200@172.22.1.103:5060>
Contact: <sip:100@172.22.1.65>
Call-ID: 29f646f372613c664214ca2e37c35f67@172.22.1.65
CSeq: 102 INVITE
User-Agent: Asterisk PBX
Max-Forwards: 70
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY
...
  
```

### **<příkaz> <parametr Request-URI> SIP/2.0**

Je to první řádek žádosti. V tomto parametru se nastavuje adresa SIP volaného nebo při registraci adresa SIP účtu, na který se přihlašujeme. Výchozí nastavení tohoto parametru může být podle pole To.

### **Pole Via**

Hlavička SIP obsahuje jedno nebo více polí Via, které značí celou cestu, přes kterou žádost šla. V případě odpovědi by se pole Via mělo využít ke zpětné reprodukci cesty. Součástí Via pole je parametr „branch“, který obsahuje jednoznačný identifikátor transakce (bude probráno později), a je používán proxy servery k detekci smyček. Obsah toho pole musí začínat řetězcem „z9hG4bK“. Via pole obsahuje hodnoty složené z názvu transportního protokolu použitého pro přenos zprávy, doménové jméno nebo IP adresu, a také port, na který je možno zaslat odpověď.

### **Pole From a To**

Následují dvě pole From a To. Pole From identifikuje volajícího uživatele a pole To volaného uživatele. Toto pole se skládá z adresy SIP, která je uvedena mezi znaky „<“ a „>“. Dále může obsahovat jméno uživatele uvedené v uvozovkách. Dále obě pole obsahují parametr tag, který identifikuje dialog (bude probráno později).

### **Pole Call-ID**

Pole je Call-ID obsahuje identifikátor dialogu. Hodnota v tomto poli má za cíl identifikovat všechny zprávy, které patří jednomu volání, a musí být stejná po celou dobu dialogu. Tento identifikátor se obvykle skládá z náhodného řetězce následovaného zavináčem a doménového jména nebo IP adresy volajícího.

### **Pole CSeq**

Vzhledem k tomu, že zprávy mohou procházet přes nespolehlivým přenosem, tak hlavička také obsahuje pole CSeq, která obsahuje pořadí požadavku. Pole CSeq musí zpráva obsahovat, aby mohl příjemce poznat, zda nedošlo k opakování přenosu. Toto pole kromě čísla požadavku obsahuje také příkaz, který odpovídá příkazu v žádosti.

**Pole Contact**

Pole Contact obsahuje adresu SIP volajícího a slouží pro přímé kontaktování UA. Toto pole musí být obsaženo vždy a obsahovat právě jednu adresu SIP v požadavku, kterým může začínat dialog. Tato adresa musí být platná pokaždé, když je použita v žádosti mimo dialog.

**Pole Allow**

Pole Allow obsahuje seznam příkazů podporovaných UA, které generovalo zprávu. Všechny příkazy zahrnující ACK a CANCEL, kterým jsou schopny porozumět UA, musí být zahrnuty v tomto poli. Pokud toto pole chybí, tak to neznámá, že UA, která posílá tuto zprávu, nepodporuje žádné metody. Používáním tohoto pole lze docílit snížení počtu zpráv s příkazem OPTIONS.

**Pole Content-Length**

Toto pole značí délku těla zprávy. Je to číslo v desítkové soustavě a znamená počet bytů. Aplikace může použít toto pole k použití velikosti těla zprávy. Pokud je použitý přenosový protokol TCP, tak toto pole musí být použito. Jakákoliv hodnota větší nebo rovna nule je platná. Pokud zpráva neobsahuje tělo musí tato hodnota být nastavena na nulu.

**Pole Content-Type**

Pole Content-Type obsahuje typ těla zprávy poslané příjemci. Pokud je délka těla nenulová, tak je toto pole povinné.

**Pole Max-Forwards**

Pole slouží k nastavení maximálního počtu skoků, kterou zpráva projde na cestě od odesílatele k příjemci. Pole obsahuje celočíselnou hodnotu, která se zmenšuje o jedničku s každým krokem. Pokud pole obsahuje nulu předtím než zpráva dosáhne cíle, tak je zamítnuta s chybovým kódem 483 (Too Many Hops).

**Pole Record-Route**

Zpráva může těchto polí obsahovat více. Toto pole vkládají proxy servery do žádostí, aby zajistily, že odpověď půjde stejnou cestou zpátky.

### **Pole User-Agent**

Toto pole označuje informaci o použitém UA klientovi a jeho verzi.

## **2.3.4 Transakce**

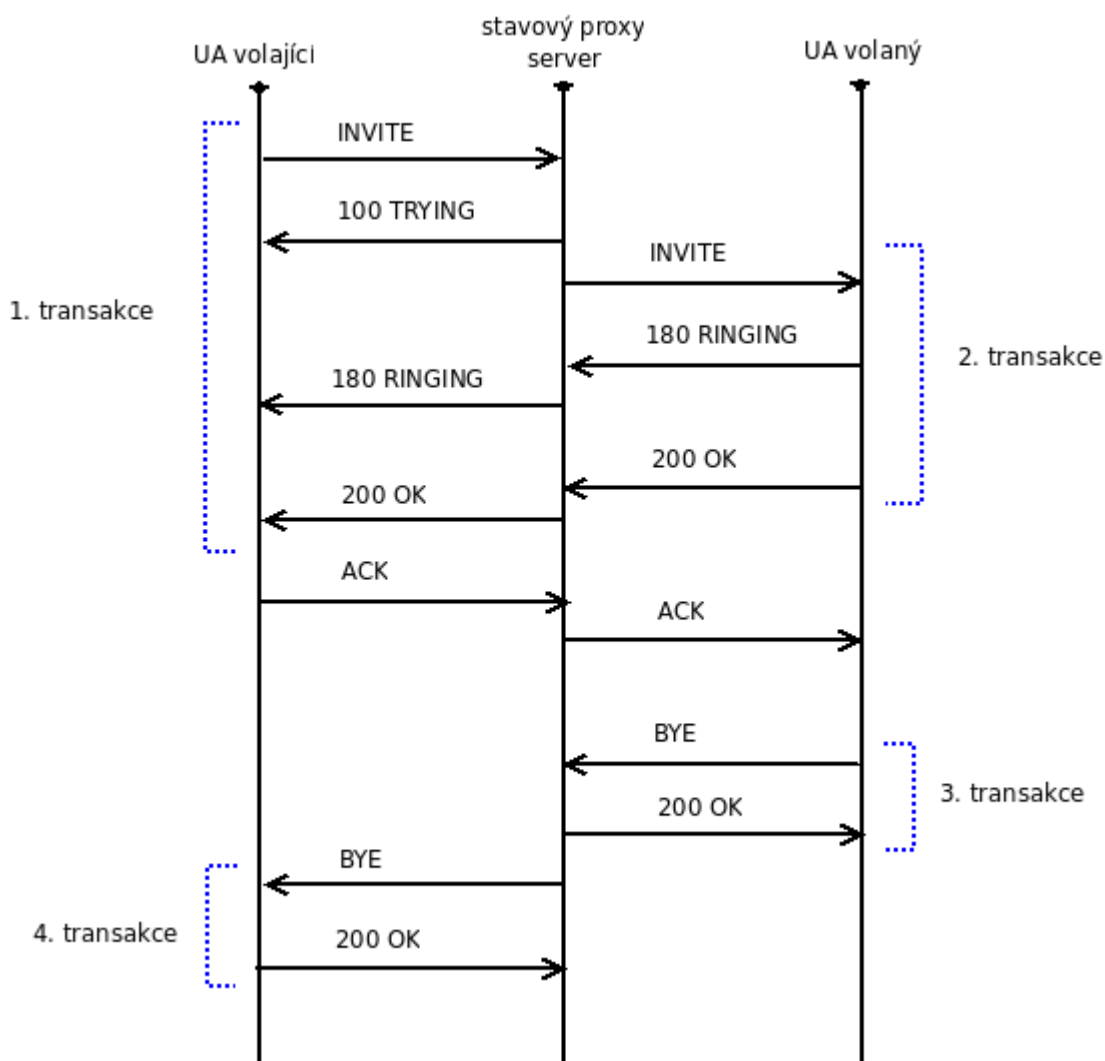
Transakce je výměna informací mezi dvěma entitami, kde informace v rámci jedné transakce spolu souvisí. I když jsou zprávy SIP posílány nezávisle SIP je transakční protokol. Znamená to, že komunikace se skládá z nezávislých výměn jednotlivých zpráv. Konkrétně transakce SIP obsahuje jednu zprávu typu žádost a několik odpovědí na tuto žádost. Tyto odpovědi zahrnují žádnou nebo více dočasných odpovědí (s návratovým kódem 1xx) a jednu nebo více konečných odpovědí (s jiným návratovým kódem než 1xx). Například žádost s INVITE se může větvit v proxy serveru a tudíž může klient dostat více konečných odpovědí.

V případě, že žádost obsahovala příkaz INVITE, tak transakce obsahuje i zprávu s ACK pouze tehdy, jestliže konečná odpověď neobsahovala návratový kód 2xx. Důvodem k tomuto kroku je důležitost doručení všech odpovědí 200 (OK) k UAC na příkaz INVITE. Za doručení a opakování přenosu všech těchto odpovědí k UAC má zodpovědnost UAS a za potvrzení s příkazem ACK má zodpovědnost UAC. Od poslání příkazu s ACK je opakování přenosu pouze na UAC, proto je efektivnější považovat to za další transakci. V případě, že volající UAC nepošle zprávu s příkazem ACK nebo se ztratí na cestě, tak volaný UAS bude opakovat posílání zprávy s kódem 200 (OK).

Transakce mají jak klientská strany, tak serverová strana. Klientská strana v transakci pošle žádost a serverová strana posílá odpovědi. Tyto transakce jsou logické funkce a existují v UA a stateful proxy serverech. Stateless proxy servery neobsahují transakce, protože pouze přeposílají zprávy. Způsob identifikace transakce je v hlavičce SIP v poli Via paramtrem „branch“ nebo pomocí číselné hodnoty CSeq. Vzhledem k tomu, že transakce probíhají sekvenčně, tak zařízení může mít v daný moment pouze jednu transakci k jinému konkrétnímu zařízení.

Na následujícím obrázku lze vidět, jak při volání přes proxy server vznikne první transakce při žádosti o spojení (příkaz INVITE) mezi UA volajícího a stateful proxy serverem na jedné straně a druhá transakce mezi tímto serverem a volaným UA na straně druhé. Obdobně při ukončení hovoru (příkaz BYE) vznikne třetí transakce mezi volaným UA a stateful proxy serverem a

čtvrtá transakce mezi tímto serverem a volajícím UA. Žádost s příkazem ACK se neuvádí, že je v transakci, protože transakce musí mít alespoň jednu konečnou odpověď, což zde není splněno.



Obrázek 6: Transakce SIP

### 2.3.5 Dialog

Na předchozím obrázku byly ukázané transakce. První transakce obsahuje zprávu s příkazem INVITE a čtvrtá transakce zprávy s příkazem BYE, protože jsou tyto transakce mezi stejnými zařízeními (UA volajícího a proxy serverem) během jednoho hovoru, tak tvoří dialog. Podobně druhá a třetí transakce tvoří dialog, protože jsou také mezi stejnými zařízeními (proxy serverem a UA volaného). Dialog tedy reprezentuje peer-to-peer SIP spojení mezi dvěma UA, které trvá nějakou dobu. Dialog tudíž napomáhá k seřazení zpráv mezi UA a odpovídajícími žádostmi

mezi nimi. K tomu se využívá pole CSeq z hlavičky SIP obsahující číslo, které je stejné v rámci transakce, a tak seřazuje transakce uvnitř dialogu. To ovšem znamená, že uvnitř dialogu může být současně v každém směru pouze jedna transakce. Lze to chápat také tak, že dialog je posloupnost transakcí.

Dialog je identifikován každým UA identifikátorem, který je složen z Call-ID hodnoty, lokálního parametru tag a cílového parametru tag. Tyto parametry jsou z polí From a To v hlavičce SIP. Identifikátor dialogu mezi dvěma UA není stejný. Lokální parametr tag na jednom UA je cílový parametr tag na druhém UA. Tyto parametry pomáhají generovat unikátní identifikátor dialogu.

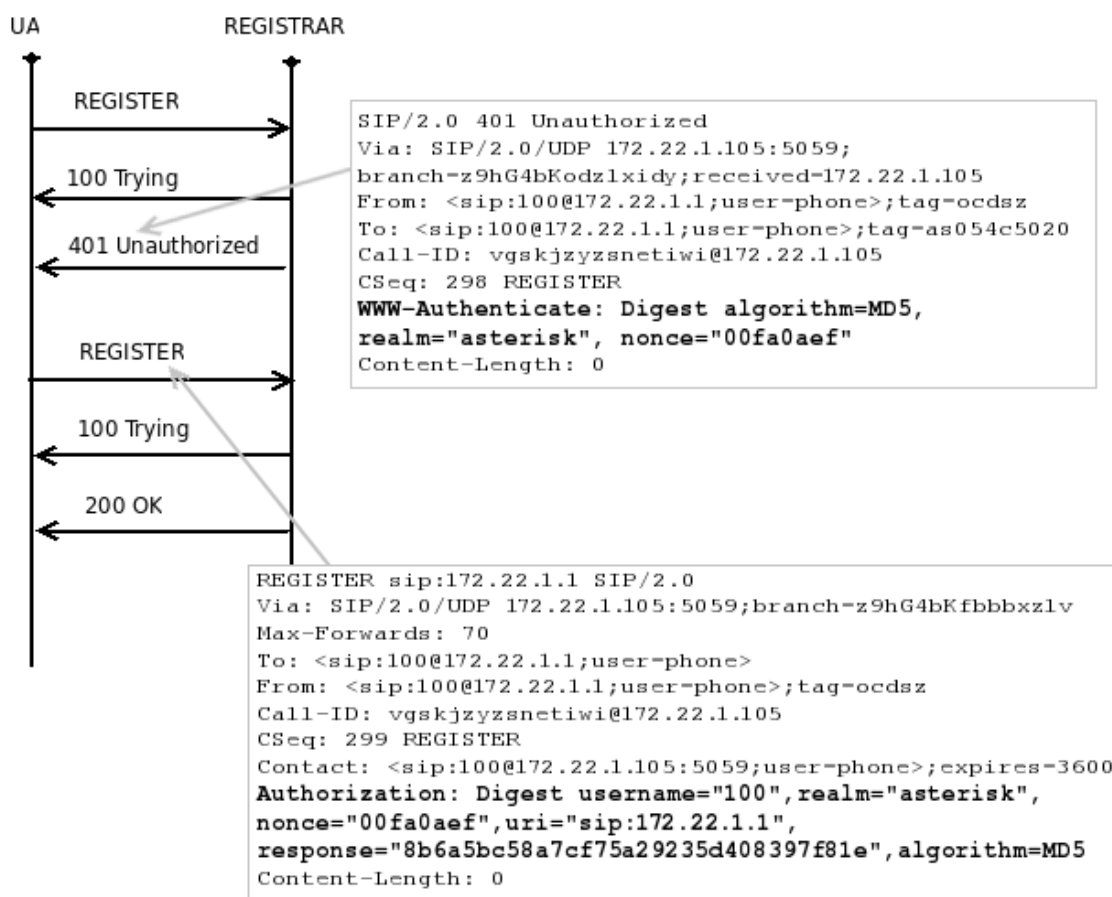
### **Ukázka hlavičky SIP s identifikátorem transakce**

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP
172.22.1.103:5060;branch=z9hG4bK349111469;received=172.22.1.103;rport=5060
From: "200" <sip:200@172.22.1.1>;tag=852125087
To: "200" <sip:200@172.22.1.1>;tag=as2e273ddc
Call-ID: 1765981028@172.22.1.103
CSeq: 7625 REGISTER
User-Agent: Asterisk PBX
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY
Expires: 60
Contact: <sip:200@172.22.1.103:5060>;expires=60
Content-Length: 0
```

## **2.3.6 Registrace a autentizace uživatelů**

Registrace znamená, že UA oznámí registrar serveru, na jaké IP adrese a portu se nachází. Autentizace znamená, že se ověří uživatelské jméno a heslo.

Registrace probíhá tak, že klient pošle žádost s REGISTER a server odpoví 200 OK, pokud je registrace úspěšná. V případě, že server vyžaduje autentizaci, tak klient dostane odpověď s kódem 407 Proxy Authentication Required nebo 401 Unauthorized. V těchto odpovědích jsou v hlavičce SIP pole WWW-Authenticate nebo Proxy-Authenticate a obsahují výzvu. Klient následně vytvoří novou žádost s příkazem REGISTER a vloží do ní pole Authorization nebo Proxy-Authorization s výslednými údaji. Na to server odpoví zprávou s kódem 200 (OK) v případě, že jsou údaje správné. Tato situace je vidět na následujícím obrázku.



Obrázek 7: Registrace klienta na SIP server

## 2.4 Směrování

Uvažujme typickou situaci, že uživatel sip:a@a.com chce volat uživateli sip:b@b.com. Ani jedna z těchto adres neříká nic o poloze uživatelů. Uživatel sip:a@a.com proto pošle žádost INVITE na svůj proxy server. Tento proxy server žádost pošle dál, až se dostane na proxy server, který zná polohu uživatele sip:b@b.com. Tento proces se nazývá směrování. UA uživatele sip:b@b.com posle odpověď volajícímu. V této odpovědi je v hlavičce SIP pole „Contact“, které se využívá pro přímé kontaktování mezi UA.

Přímé kontaktování je výhodné z hlediska rychlosti, protože zprávy jsou doručovány přímo a není zatěžován proxy server. Navíc proxy servery obsahují většinou složitější směrování. Přímé kontaktování se může využít např. pro zahájení nebo ukončení hovoru. V případě ukončení



jedna z UA, pošle žádost s BYE přímo na druhé UA podle pole „Contact“. Bohužel existují i nevýhody a jsou zejména špatná průchodnost skrz NAT. Nebo pokud proxy server zajišťuje účtování hovorů, tak potřebuje být informován o ukončení hovoru. Proto existuje možnost, jak informovat UA, že chceme, aby posílal všechny zprávy o spojení přes proxy server. V žádosti se zaznamenává cesta do pole v hlavičce SIP „Record-route“, v případě více serverů se toto pole opakuje s každým z nich. Odpověď jde potom směrována po této zaznamenané cestě.

## 3 Asterisk

Cílem této kapitoly je popsat použití softwaru Asterisk jako SIP proxy server. A dále ukázat možnosti konfigurace pro připojení klientů a následné možnosti využít tento server pro připojení do dosavadní IP telefonie FIT.

Asterisk [6] je kompletní softwarová PBX (private branch exchange – pobočková ústředna). Umí komunikovat pomocí protokolu SIP, IAX i H.323 [11]. Podpora protokolu H.323 je za pomoci modulu z projektu OpenH323 [12]. Asterisk vyvíjí americká firma Digium, Inc. (<http://www.digium.com/>). Firma investuje do zdrojového kódu a také do telefonního hardware, který pracuje s Asteriskem. Kód je distribuován jako open-source pro platformy Linux a BSD pod licencí GNU. Výjimku tvoří spojení s projektem OpenH323 a to kvůli dostupnosti podpory pro H323. Za pomoci Asterisku jsme schopni vytvořit most mezi klasickou telefonní sítí PSTN, VoIP operátory a softwarovými a hardwarovými SIP klienty. Asterisk je kombinace všech typů proxy serverů uvedených ve 2. kapitole. To znamená, že může sloužit jako proxy, registrar a redirect server.

Asterisk nepotřebuje žádný další hardware pro VoIP, pokud není připojen do sítě PSTN. Pro odchozí/příchozí hovory může být použito několik VoIP operátorů. Pro spojení s digitální nebo analogovou ústřednou má Asterisk mnoho ovladačů pro různé hardwarové zařízení, nejčastěji pro zařízení od výrobce Digium. Hardwarové zařízení jsou telefonní PCI karty, které jsou v současné době cenově dostupné a stojí podle počtu portů pro telefonní linky. Např. PCI karta (bez modulu) pro čtyři moduly stojí cca 2700,- Kč a jednotlivé moduly stojí cca 2500,- Kč. Více na <http://www.voipobchod.cz/>.

### 3.1 Služby ústředny

Tak jako klasické pobočkové ústředny nabízí i Asterisk nabízet určité služby. Díky tomu, že je to software, můžeme ho libovolně rozšiřovat a nabízet mnohem více než klasické ústředny. Základní služby Asterisku jsou:

- hlasová schránka – Zaznamená hovor, pokud uživatel není přihlášený. Tento hovor se uloží ve formátu „wav“ a pošle emailem uživateli nebo si jej může vyzvednout po zavolání do hlasové schránky a zadání hesla.
- konferenční hovory – Tyto hovory se vytvářejí tak, že jednotliví účastníci vytočí speciální číslo, které je určeno pro konferenční hovor. Tímto se vytvoří konferenční hovor a účastníci navzájem spojí.
- IVR (Interactive voice response) – interaktivní hlasový průvodce, který se používá v call-centrech, jako jsou například linky mobilního operátora, bankovníctví přes telefon, apod.. Tato funkce navádí volajícího, jak postupovat pomocí předem připravených audionahrávek a umožňuje detekovat stisknuté tóny kláves během normálního hovoru.
- fronta hovorů – Používá se, pokud vytočíme uživatele a ten má zrovna hovor, tak se dostaneme do fronty a čekáme až dotelefonuje a následně se spojí náš hovor.
- účtování hovorů – U každého hovoru umožňuje zaznamenat informace o jeho délce, účastnících, ceně a další.
- identifikace volajícího – Tato funkce zobrazí jméno uživatele, který volá.
- překlad čísel ENUM – Pomocí této funkce můžeme přímo v Asterisku využívat překladu telefonních čísel na adresy SIP. Tato funkce sama provede překlad telefonního čísla pomocí DNS systému.

## 3.2 Vnitřní architektura

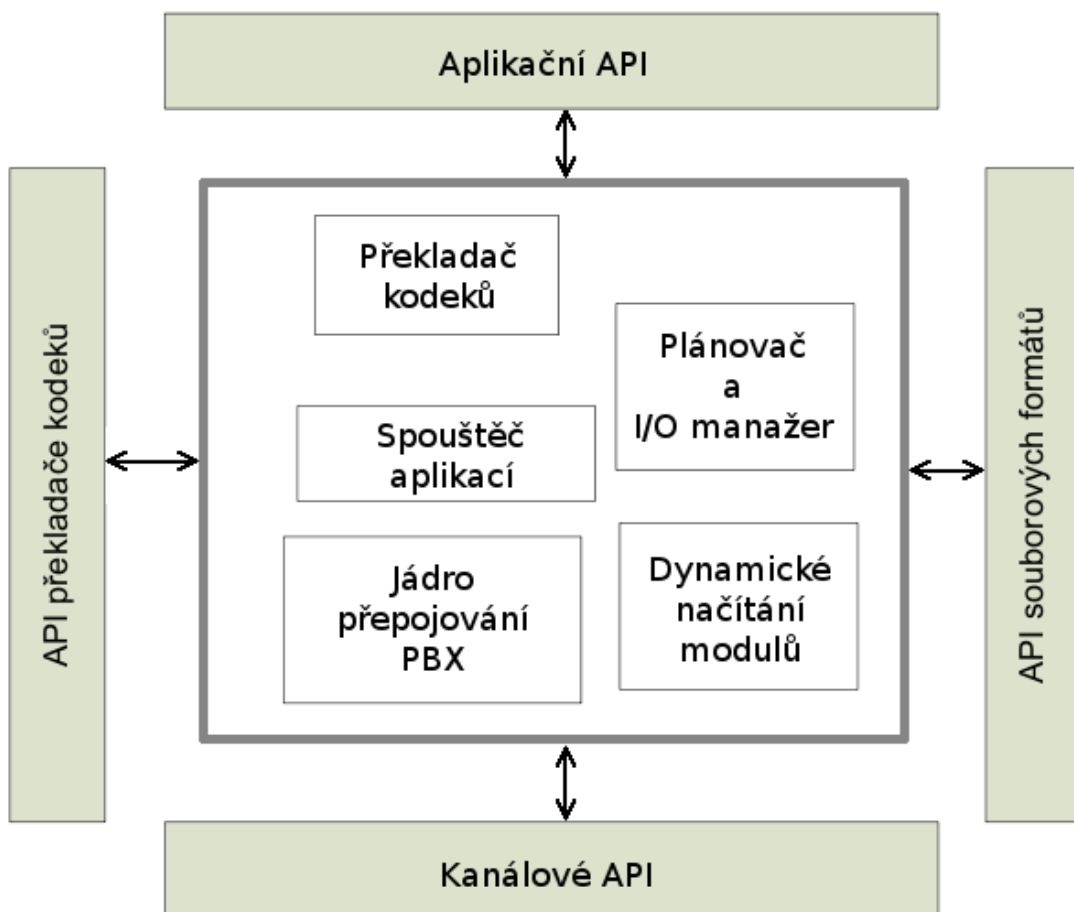
Asterisk je navržen pro maximální flexibilitu. Asterisk má pokročilé vnitřní systémové jádro. Toto jádro ovládá vnitřní komunikaci mezi specifickými signalizačními protokoly, hlasovými kodeky a různorodým hardwarem, jako jsou různé telefonní karty. Toto vše dovoluje Asterisku použít vhodných technologií, které jsou dostupné teď nebo v budoucnu a které zajistí nezbytnou funkčnost spojení hardwaru a softwarových aplikací.

Na následujícím obrázku jsou znázorněny vnitřní části ovládané jádrem systému Asterisk a jeho vnější rozhraní. Vnitřní části jsou následující:

- *jádro přepojování PBX (PBX Switching)* – Je to podstatná část systému Asterisk – systém přepínání pobočkové ústředny. Tato část se stará o spojování hovorů mezi různými uživateli a automatizovanými úlohami (IVR, hlasová schránka, atd.). Jádro

přepojování transparentně spojí hovory přicházející na různé hardwarové a softwarové rozhraní.

- *spouštěč aplikací (Application Launcher)* – Spouští aplikace, které zajišťují služby pro uživatele, jako jsou hlasová schránka, přehrávání souborů, atd.
- *překladač kodeků (Codec Translator)* – Tato část používá moduly kodeků pro kódování a dekódování různých audio formátů použitých v telefonním prostředí.
- *plánovač a I/O manažer (Scheduler and I/O Manager)* – Ovládá nízkourovňové úlohy plánování a systémového řízení pro optimální výkon.



Obrázek 8: Architektura serveru Asterisk

Dále je na obrázku znázorněno modulové rozhraní API, které se používá pro načtené moduly, podporovaný hardware a další. Použitím těchto modulů se jádro Asterisku nemusí starat o detaily spojení volajícího, použitý typ kodeku atd. API jsou následující:

- *Kanálové API* – Tato část pracuje s příchozím spojením typu VoIP, ISDN a některými dalšími. Dynamické moduly jsou načteny a udržují informace o detailech v nižší vrstvě.
- *Aplikační API* – Díky této části určité moduly běží a posílají různé úkoly, aby splnily určité funkce. Konferenční hovory, zasílání zpráv, hlasová schránka, přímý přenos dat a další úkoly, které by PBX systém měl vykonat teď nebo v budoucnu jsou ovládány těmito oddělenými moduly.
- *API překladače kodeků* – Načítá moduly na podporu různých kodeků (GSM, Alaw, ...)
- *API souborových formátů* – Pomocí této části se čte nebo zapisuje v různých formátech v souborovém systému.

Používáním těchto API Asterisk můžeme použít funkce jádra systému na různých technologiích existujících v telefonní oblasti. Modularita je to, co Asterisku dovoluje hladce integrovat současný telefonní hardware a technologie založené na VoIP. Schopnost načíst modul kodeku dovoluje Asterisku podporovat jednak extrémně úsporný kodek s velmi pomalým připojením jako je telefonní modem, tak i poskytovat vysokou audio kvalitu přes kvalitní připojení.

V následující tabulce jsou uvedeny kodeky, které Asterisk podporuje. MOS (Mean Opinion Score) znamená kvalitu hovoru, kde číslice pět je nejlepší a číslice jedna nejhorší.

Standard	Algoritmus	Rychlost [kbps]	MOS
G.711	PCM	64	4,1
G.726	ADPCM	32	3,85
GSM	RPE-LTP	13	3,5
G.729	CS-ACELP	8	3,92
G.723.1	MP-MLQ	6,3	3,9
G.723.1	ACELP	5,3	3,64

*Tabulka 1: Souhrn hlasových kodeků*

V následující tabulce je uveden přehled základních konfiguračních souborů. Všechny soubory jsou umístěny v adresáři /etc/asterisk.

Název souboru	Popis souboru
extensions.conf	Konfigurace číslovacího plánu.
sip.conf	Konfigurace protokolu SIP a klientů SIP.
rtp.conf	Konfigurace protokolu RTP.

*Tabulka 2: Konfigurační soubory serveru Asterisk*

### 3.3 Číslovací plán (Dial plan)

Číslovací plán je důležitá část systému Asterisk. Definuje, jak směřovat příchozí a odchozí hovory. Je složen z posloupnosti příkazů nebo kroků, které Asterisk postupně vykonává. Na rozdíl od tradičních telefonních systémů je číslovací plán plně přizpůsobitelný. Abychom mohli úspěšně nastavit systém Asterisk, tak potřebujeme porozumět číslovacímu plánu.

Soubor „extensions.conf“ konfigurace Asterisku slouží pro definici číslovacího plánu. Obsah souboru je organizován do sekcí, které mohou být staticky nastaveny a definovány. Speciální sekce jsou „general“ a „globals“. Sekce a název kontextu se uvádí v hranatých závorkách, za kterými následují volby a definice.

Kontext je sekce, kde definujeme jednotlivé telefonní linky, tj. část číslovacího plánu. Kontextů může být více a je dobré je tvořit kvůli přehlednosti. Pomocí nich můžeme definovat zcela jiný číslovací plán pro různé domény, na které se klienti přihlašují. Speciální typy kontextů jsou makra, kde začátek jména musí být „macro-“. Když Asterisk přijímá hovor na kanálu, podívá se do kontextu definovaného tomuto kanál pro příkaz, co má dělat. Kontext definuje různé příkazy v závislosti na vytáčené lince.

Tak jako tradiční ústředny přiřazují linkám telefony, menu apod., tak v Asterisku definujeme seznam příkazů, které se spustí pro danou linku. Příkazy jsou spuštěny v pořadí definovaném podle jejich priority. Některé příkazy (Dial nebo GotoIf) mají schopnost přesměrovat provádění příkazů jinam podle zadané podmínky.

Definice linek se zadává ve tvaru:

*exten = linka, priorita, příkaz(parametry)*

popis voleb:

- *linka* – Zde se uvádí číslo linky, které se může být tvořeno znaky a číslicemi. Pokud jako první znak uvedeme „\_“ pak to znamená, že některé znaky mají speciální význam.

Speciální znaky:

- X – číslice 0 až 9
- Z – číslice 1 až 9

- N – číslice 2 až 9
- [0145] – jakákoliv číslice ze závorky
- . - tečka znamená jakékoliv číslo (i různě dlouhé)

Např.: `_NXXXXXXXX` – Znamená číslo začínající 2 až 9 a následuje libovolných 8 číslic.

- *priorita* – Pro jednu linku může být definováno více kroků to se provádí tak, že se každý krok je definován na samostatném řádku se stejnou linkou a číslem priority. Začíná se prioritou 1, s tím, že každá následující je o 1 větší. Může se uvést `n` (jako next), které znamená aktuální priorita plus 1, nebo `s` (jako same), která znamená aktuální priorita.
- *příkaz* – Typicky `Dial(SIP/uživatelské jméno)` pro vytočení nebo `Hangup` pro ukončení hovoru.

Pozn.: Asterisk v dokumentaci nedoporučuje používat „`_.`“ (podtržítka a tečka pro libovolné číslo), protože to může vést k neočekávaným výsledkům. Lepší je použít „`_X.`“ (podtržítka X a tečka).

Příklad: `exten => _0,1,Dial(SIP/${EXTEN:1}@xxxxxxxx)`

Tento příklad ukazuje typické volání přes nulu do vnější sítě. „`_0.`“ značí libovolné číslo začínající nulou. „`Dial`“ je příkaz pro vytočení, který má jako vnitřní parametr identifikátor volajícího. Identifikátor je složený z kanálu (zde SIP) za ním „`/`“ následuje číslo linky nebo číslo linky `@` název uživatele (uvedeného v `sip.conf` volba register), pokud je hovor přeměřován na našeho VoIP operátora. Zde je uvedena proměnná `${EXTEN}`, která značí právě volané číslo. Proměnná také obsahuje „`:`“ jako oddělovač a následuje číslice, která značí počet vynechaných číslic. V tomto případě je výsledkem volané číslo bez první číslice.

## 3.4 Architektura SIP v Asterisku

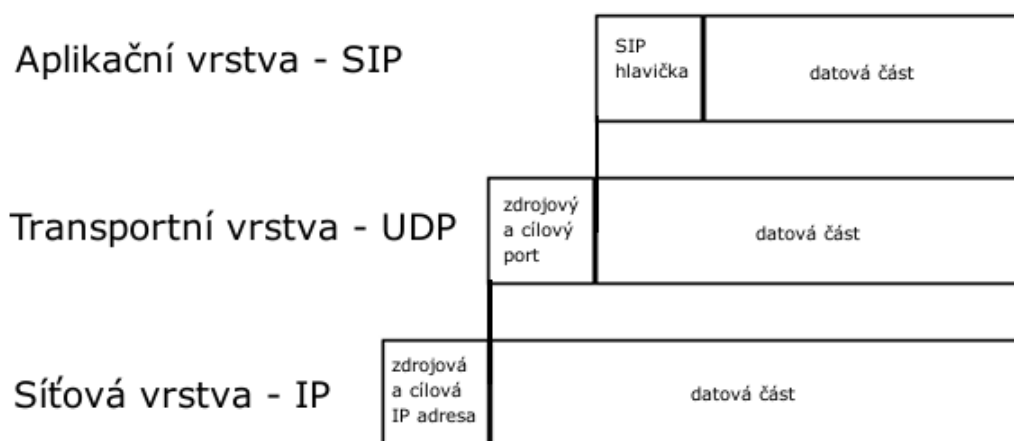
V Asterisku musíme nadefinovat klienty, abychom se na ně později mohli odkázat v konfiguraci číslovacího plánu. Asterisk má vnitřně rozdělené klienty na dva typy objektů:

- „`peer`“ – Je to typ klienta, na který Asterisk posílá hovory. Typicky se využívá jako účet pro VoIP operátora, přes kterého voláme, ale který nám nepřidělil příchozí adresu. Tohoto se využívá při peeringu (vzájemné propojení telefonní sítě dvou operátorů).
- „`user`“ – Je to typ klienta, který může volat přes Asterisk. Typicky nějaký telefon, na který nelze volat.

- „friend“ – Je to kombinace obou předchozích typů. Typicky koncové zařízení, VoIP operátor. U tohoto typu si Asterisk vytvoří v paměti oba předchozí typy klientů.

### 3.5 SIP a NAT v Asterisku

NAT se používá pro připojení více počítačů do sítě internet, když máme k dispozici pouze jednu IP adresu. V takovém případě potřebujeme router, který bude provádět překlad NAT. Tento router připojí naši lokální síť (kde jsou použity IP adresy z neveřejného rozsahu) do sítě internet. Nevýhodou takového připojení je to, že se nelze na PC uvnitř lokální sítě připojit z internetu. Lze to obejít pomocí přesměrování portu, ale u SIP protokolu to přesto způsobuje komplikace. Je to proto, že IP adresa je uvedena i v datové části UDP datagramu (v hlavičce SIP pole Contact) a nejen v hlavičce IP.



Obrázek 9: Struktura paketu

#### Ukázka hlavičky SIP

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.22.1.103:5060;branch=z9hG4bK349111469;rport=5060
From: "200" <sip:200@172.22.1.1>;tag=852125087
To: "200" <sip:200@172.22.1.1>;tag=as2e273ddc
Call-ID: 1765981028@172.22.1.103
CSeq: 7625 REGISTER
Contact: <sip:200@172.22.1.103:5060>;expires=60
Content-Length: 0
```



### 3.5.1 Server za NATem

Může se stát, že máme server připojený k internetu a ISP nám sice přidělil veřejnou IP adresu, ale tato adresa je za NATem. Pokud se jedná o NAT 1:1 (jedna veřejná IP adresa je mapovaná na jednu privátní IP adresu) stačí nastavit volby v konfiguraci sip.conf v sekci globals dle následující tabulky:

Volba	Popis
externip = 1.2.3.4	Veřejná IP adresa, která se uvádí ve zprávách SIP.
externhost = asterisk.example.com externrefresh = 10	DNS jméno, které směřuje na veřejnou IP adresu. Určuje jak často se bude překládat DNS jméno na IP adresu.

*Tabulka 3: Konfigurace NATu v souboru sip.conf*

V případě NATu 1:n (jedna veřejná IP adresa je mapovaná na více privátních IP adres) jsou volby stejné, ale musí se ještě na routeru nastavit přeposílání portů. V případě linuxového routeru to lze udělat pomocí „iptables“. Porty k přesměrování jsou 5060 UDP pro SIP a 10000 až 20000 UDP pro hlasové pakety (RTP stream).

#### **Volba localnet**

Klienti zapadající do síťových rozsahů definovaných pomocí volby localnet budou vyřazeni z použití NATu Asteriskem. Výsledkem bude, že jako zdrojová IP adresa s žádostí SIP nebo odpovědí se použije interní IP adresa síťového rozhraní. Tato volba je důležitá, pokud máme klienty uvnitř vnitřní sítě (bez NATu), tak i venku (NAT u serveru Asterisk).

```
localnet=192.168.0.0/255.255.0.0
```

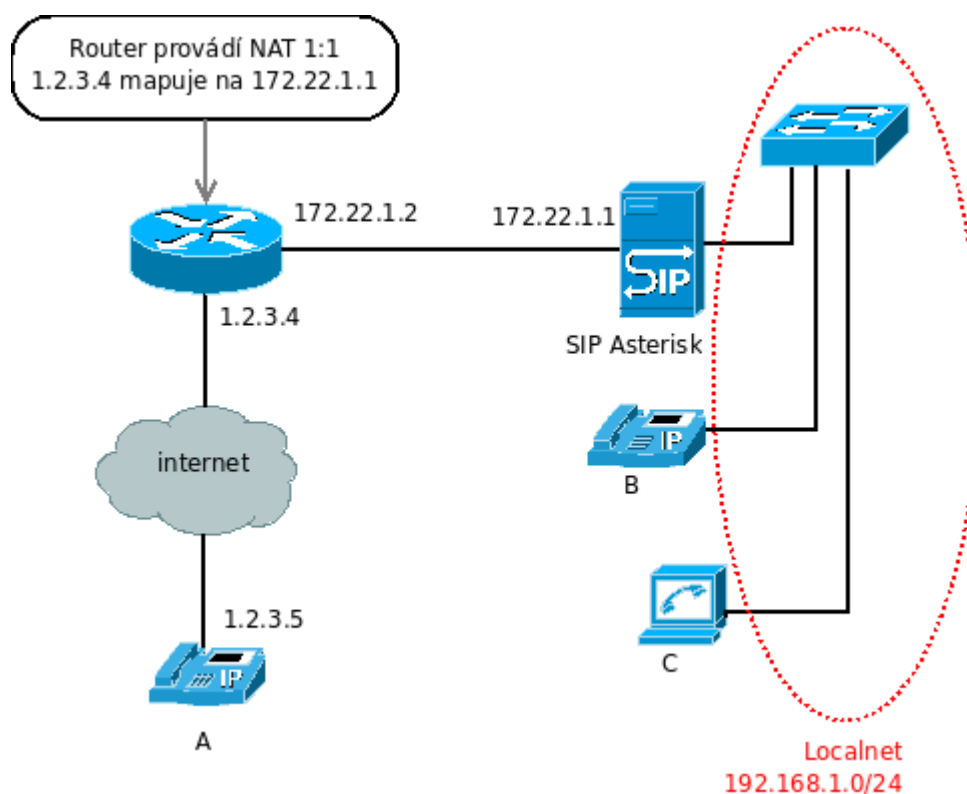
```
localnet=10.0.0.0/255.0.0.0
```

```
localnet=172.16.0.0/12
```

```
localnet=169.254.0.0/255.255.0.0
```

Na následujícím obrázku je vidět příklad. Je zde SIP server Asterisk, připojený do internetu přes NAT 1:1 a do lokální sítě. Klienti se mohou připojit na SIP server Asterisk z internetu a lokální sítě. Kvůli klientům z internetu musíme nadefinovat externí IP adresu na 1.2.3.4 pomocí volby „externip“. Nyní Asterisk bude ve všech zprávách SIP posílat veřejnou IP adresu 1.2.3.4, jak pro klienta A z internetu, tak pro klienty B a C z lokální sítě „Localnet“. Protože chceme,

aby klientům B a C posílal Asterisk ve zprávách SIP IP adresu rozhraní připojeného do této lokální sítě, tak musíme nastavit volbu localnet na 192.168.1.0/24.



Obrázek 10: Připojení serveru Asterisk za NATem

### 3.5.2 Klient za NATem

Pokud je klient za NATem, tak router provádí překlad a tím mění IP adresu. Ale router ji mění pouze v IP paketech a už ne ve zprávách SIP. Potom zde nastává problém, jednak v přenosu zpráv SIP a také v přenosu hlasu (RTP stream).

#### Volba NAT

Pro protokol SIP je definováno rozšíření RFC3581 [13], které pomáhá s průchodem přes NAT. Pokud je v hlavičce SIP v poli Via parametr rport, tak má druhá strana poslat odpověď na adresu odkud přišla, a ne na tu, která je uvedena v hlavičce SIP.

možnosti volby „nat“:

- „yes“ – Vždy posílá pakety odkud přišly, bez ohledu na to co je v hlavičce SIP.
- „no“ – Odešle pakety odkud přišly, pokud je v hlavičce SIP v poli Via parametr „rport“

- „never“ – Vždy posílá pakety podle toho, co je v hlavičce SIP.
- „route“ – Předpokládá, že klient je za NATem, ale neposílá parametr „rport“ v hlavičce SIP v poli Via.

### **Volba qualify**

Pokud je volba zapnutá, tak Asterisk posílá zprávy SIP s příkazem OPTIONS každých 60 sekund, aby zkontroloval zda je zařízení stále aktivní. Pokud zařízení neodpoví, tak Asterisk předpokládá, že zařízení je neaktivní. Tato volba se také používá pro udržení otevřené UDP relace k zařízení, které je za NATem.

možnosti volby „qualify“:

- „no“ – vypnutí volby
- „yes“ – zapnutí volby, výchozí časový limit je dvě sekundy
- „xxx“ – zapnutí volby a nastavení časového limitu na hodnotu xxx.

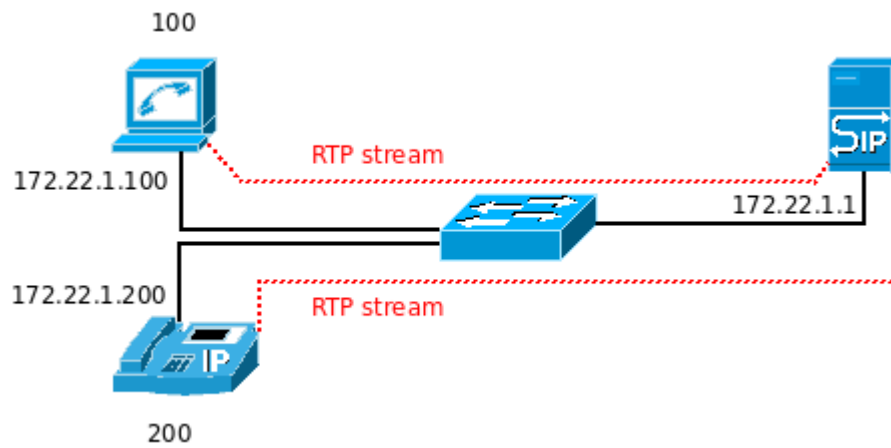
Asterisk posílá zprávy SIP s příkazem OPTIONS každých šedesát sekund a tato hodnota je neměnná. Hodnota qualify = xxx znamená časový limit (timeout) potom, co je paket poslán, ale předtím, než je koncové zařízení označeno Asteriskem jako neaktivní. Jestliže zařízení neodpoví Asterisk to zkouší celkem šestkrát. Pokud ani po té nedostane odpověď zkouší to znovu až nastane šedesátisekundový cyklus. Všechny tyto hodnoty (mimo časového limitu) jsou definovány přímo ve zdrojovém kódu.

### **Volba canreinvite**

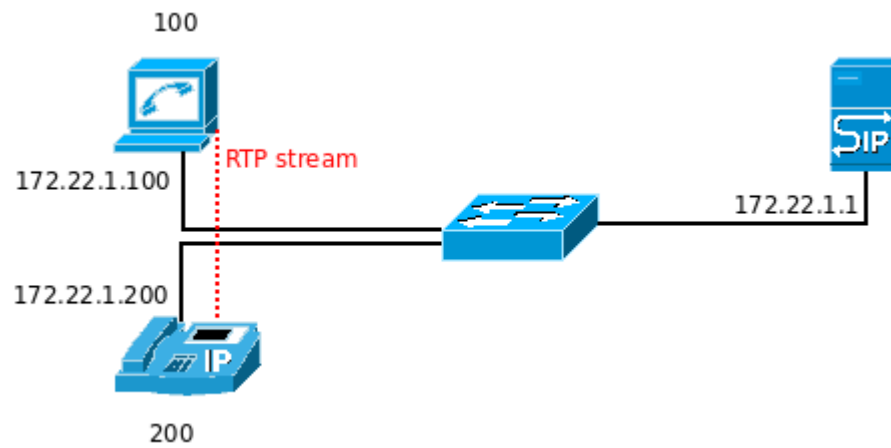
Pokud chceme, aby mezi klienty procházel přenos hlasu přímo a server Asterisk sloužil jen jako signalizační server, tak nastává problém. Pokud jsou oba klienti za NATem, tak se nemohou navzájem kontaktovat. Proto využijeme volbu canreinvite v konfiguraci sipu. Tato volba způsobí, že pokud klient A vytočí klienta B, tak server Asterisk přijme hovor od klienta A a vytvoří nový hovor ke klientu B. Vzniknou tedy dva dialogy a přenos hlasu bude procházet přes server Asterisk.

možnosti volby „canreinvite“:

- „yes“ – vytvoří RTP stream, přímo k druhému klientovi
- „no“ – vytvoří RTP stream přes asterisk, zde jde fakticky o dva hovory a mohou být zde použity různé kodeky



Obrázek 11: Volba canreinvite s hodnotou „no“



Obrázek 12: Volba canreinvite s hodnotou „yes“

## 4 Propojení SIP serveru s dosavadní IP telefonii na FIT

V současné době zajišťuje IP telefonii pro celé VUT Brno externí firma. Telefonní síť je postavena na technologii Avaya. Tato technologie by měla být kompatibilní se SIP protokolem. Dále je telefonní síť VUT propojena se sítí CESNET. A na síť CESNET je možné se dovolat díky technologii ENUM.

Cílem této kapitoly je nakonfigurovat SIP server Asterisk, který bude komunikovat s VoIP sítí FIT. A dále, aby se na tento server mohli přihlašovat zaměstnanci FIT, kteří potom budou moci telefonovat na VUT zdarma.

Možnosti připojení jsou tedy následující:

1. Nechat si přidělit účet od Cesnetu [14] nebo VoIP operátora (např. Vox 802 firmy MATTES AD, spol. s r.o. <http://www.802.cz>) a nakonfigurovat Asterisk, aby mohli uživatelé volat přes tento účet.
2. Použít na Asterisku překlad ENUM, který zajistí volání do celé VUT sítě přes síť CESNET.

První způsob obsahuje nutnost se někde registrovat a v případě VoIP operátora platit za telefonní číslo. Většina českých VoIP operátorů spojuje hovory do VUT sítě zdarma. Protože druhá možnost je jednodušší, tak se jí nadále budu zabývat.

### 4.1 Adresování v telefonii a standard ENUM

V současnosti používáme následující typy adres pro volání podle typu sítí:

- PSTN – jedná se o klasickou telefonní síť s přepínáním okruhů, kde se používá telefonní číslo např: +420800123456
- H.323 – je to VoIP protokol, kde vypadá adresa URI např. h323:user@h323.com
- SIP – je to o moderní VoIP protokol, kde vypadá adresa URI např: sip:user@sip.com

Pokud chceme někomu zavolat, musíme znát telefonní číslo. S nástupem moderních technologií jako je VoIP vznikl problém, jak toto číslo namapovat na adresu URI. Můžeme samozřejmě v uživatelské části adresy URI uvést telefonní číslo, což se v současnosti u operátorů VoIP velmi používá. Ale je potom problém, jak určit část adresy SIP určenou doméně. Většina operátorů VoIP to řeší vlastním číselným plánem, tzn., že číslo s určitým prefixem posílá do určité sítě. Zde nastávají problémy s individuálním přenosem čísla, neboť pokud se změní směrování pro určitý prefix, tak to znamená zásah v konfiguracích na obou stranách (jak na straně vlastníka čísel, tak na straně všech kdo směřují čísla přímo k vlastníkovi).

Tyto problémy se snaží řešit ENUM [15]. Jde ve své podstatě o mapování telefonních čísel E.164 do doménových jmen. Proto lze ENUM pochopit jako most mezi telefonními čísly jednotlivých sítí a URI identifikátory, které jsou uloženy v DNS záznamech. Standard ENUM je tvořen za pomoci spolupráce dvou organizací. Je to ITU (International Telecommunication Union) a organizace IETF (Internet Engineering Task Force).

ENUM je tedy klíčovým standardem pro možnost adresovat síť založenou na IP protokolu a tradiční telefonní síti s přepínáním okruhů PSTN. Princip ENUM je jednoduchý. Jde ve své podstatě o namapování telefonního čísla podle E.164 na URI adresu, která identifikuje službu (např. SIP, H.323) nebo zdroj. Výsledkem mapování na URI adresu je telefonní číslo zapsané v opačném pořadí číslic, přičemž každá číslice je oddělena tečkou a celý řetězec je doplněn o doménu e164.arpa, která byla pro účely ENUM vytvořena. Jde tedy o podobný překlad jako v případě reverzních záznamů, kdy se IP adresa převádí na doménové jméno pomocí DNS. Např. telefonnímu číslu +420541141108 odpovídá doména 8.0.1.1.4.1.1.4.5.0.2.4.e164.arpa.

## 4.2 ENUM v ČR

Pro české telefonní čísla je k dispozici doména 0.2.4.e164.arpa, podle národní předvolby +420. Tuto doménu spravuje sdružení „cz.nic“ Registrace telefonních čísel zabezpečují registrátoři. Jsou to společnosti, které mají přístup do registru „cz.nic“ a provádí samotnou registraci telefonního čísla zákazníka. Bohužel v současnosti ne všichni operátoři podporují ENUM ve smyslu registrace vlastní domény k telefonnímu číslu. Osobně mám zkušenosti se službou VOX 802 (<http://www.802.cz>), jedná se o VoIP operátora. Je také zároveň registrátor pro všechny své telefonní čísla, ale nemá vytvořený žádný ENUM záznam, ani u něj nelze

registrovat záznam s vlastní doménou. Na jejich stránkách jsou pouze informace, že se testováním služby ENUM zabývají a všichni zákazníci budou informováni o jeho spuštění.

## 4.3 Mapování telefonního čísla pomocí ENUM

Způsob převodu telefonního čísla na URI pomocí ENUM:

1. Z telefonního čísla v mezinárodním formátu, např. +420541141108 odstraníme všechny znaky kromě číslic
2. Mezi jednotlivé číslice vložíme tečky „4.2.0.5.4.1.1.4.1.1.0.8“.
3. Zapišeme v obráceném pořadí „8.0.1.1.4.1.1.4.5.0.2.4“.
4. Tento tvar se vloží do domény e164.arpa: 8.0.1.1.4.1.1.4.5.0.2.4.e164.arpa.
5. Do systému DNS se vyše dotaz na NAPTR záznam pro příslušné doménové jméno.

Systém DNS nám vrátí požadovanou URI pro danou službu sip:420541141108@vutbr.cz

## 4.4 DNS záznam typu NAPTR

Pomocí záznamu typu NAPTR můžeme do DNS zakódovat určitou sadu pravidel. Pomocí pravidel pak můžeme URI rozložit na části, které mohou být změněny. Tento typ záznamu zahrnuje regulární výrazy, které jsou použity klientským programem k přepsání řetězce do doménového jména nebo URI adresy. Regulární výrazy jsou výhodné zejména pro malou velikost v porovnání s množstvím operací, které umožňují. Funkce přepisování řetězce podle pravidel v záznamech je užitečná v mnoha různých aplikacích.

### 4.4.1 Struktura NAPTR záznamů

NAPTR záznam obsahuje následující položky:

- Domain – Znamená doménový název, který je odkázán na tento záznam.
- TTL – Doba platnosti záznamu.
- Class – Standardní DNS pole, zde IN.
- Type – Typový záznamu, zde NAPTR.
- Order – Je 16-bitové kladné celé číslo, které specifikuje pořadí, v jakém záznamy musí být zpracovány, aby se zajistilo korektní zpracování pravidel (položka „regexp“) z těchto záznamů. Nižší číslo znamená větší prioritu. Pokud klient jednou nalezne shodu s nějakým záznamem, tak nesmí použít jiný záznam, který má menší prioritu.

- Preference – Je to 16-bitové kladné celé číslo. Toto číslo specifikuje pořadí, v jakém by měly být zpracovány záznamy, které mají stejnou prioritu (hodnotu order). Menší číslo znamená dřívější zpracování. Rozdíl oproti „order“ je, že zde může použít záznamy s vyšším číslem, ale u „order“ nesmí. Většinou se používá toto pole, abychom zvýšili preferenci určité služby oproti jiné. Např. upřednostnit službu SIP oproti službě H323.
- Flags – Toto pole obsahuje řetězec složený z příznaků, které ovládají způsob přepisování a interpretace jednotlivých polí v záznamu. Příznaky jsou písmena nebo číslice. Velikost písmen není důležitá. V současnosti se používají čtyři příznaky S, A, U a P.

Příznak	Význam
S	Výsledek se bude překládat pomocí SRV záznamu.
A	Výsledek se bude překládat pomocí A, AAAA nebo A6 záznamu.
U	Výsledek není doménové jméno, ale je to URI adresa.
P	Protokol bude určovat aplikace, která dotaz poslala.

*Tabulka 4: Hodnoty pole Flags v záznamu NAPTR*

- Service – Toto pole specifikuje typ služby a protokol. Pole je zapsané ve tvaru „protokol+sluzba“, přičemž služeb může být uvedeno více a vždy se oddělují znakem +. Toto pole může být i prázdné nebo obsahovat pouze název protokolu.
- Regexp – Řetězec obsahující nahrazující výraz, který je použitý na původní řetězec poslaný klientem, aby vytvořil doménový název, který se bude překládat v následujícím kroku.
- Replacement – Je to další název do dotazu pro NAPTR, SRV nebo adresové záznamy, podle toho jaký je zvolen příznak. Toto pole musí být doménové jméno.

### Nahrazování v pole regexp

Toto pole má následující tvar:

<oddělovač><regulární výraz><oddělovač><nahrazující výraz><oddělovač><příznaky>

kde význam je následující:

- <oddělovač> je buď „/“ nebo „!“ . Případně libovolný znak mimo číslic a znaku v příznacích jiný než „\“ . Všechny výskyty oddělovače v poli regexp musí mít stejný znak.
- <regulární výraz> je POSIX [16] rozšířený regulární výraz



- <nahrazující výraz> je řetězec obsahující alespoň jeden znak. Dále může obsahovat zpětnou referenci, která se skládá z „\“ a jedné číslice.

Klientská aplikace použije regulární výraz na originální URI (zaslané v dotazu do DNS) nebo na řetězec, který si aplikace určí. Pokud jsou v regulárním výrazu závorky, tak se nastaví zpětná reference na výsledné řetězce v těchto závorkách, která se pak použije v nahrazujícím výrazu. Nahrazující výraz je řetězec, kde se pouze nahradí zpětné reference z regulárního výrazu. Nahrazující výraz musí být po této aplikaci platným doménovým jménem nebo URI adresou podle příznaků v poli flags. Pro příznak S a A to musí být doménové jméno a pro U to musí být URI adresa.

**Například:** řetězec 541141108 a v poli regexp je „!<sup>5</sup>4114(.\*)\$!sip:\1@vutbr.cz!“

Vybereme regulární výraz <sup>5</sup>4114(.\*)\$ aplikujeme ho na řetězec 541141108 a zpětná reference „\1“ se nastaví na řetězec „1108“. V druhé části pole regexp je adresa sip:\1@vutbr.cz, kde „\1“ se nahradí řetězcem „1108“. Výsledkem tedy bude adresa SIP URI sip:1108@vutbr.cz.

## 4.4.2 Algoritmus výběru záznamu NAPTR

Režim a význam pole příznaků „flags“ a pole služeb „services“ vybere algoritmus, kde výsledek z přepsání je nový klíč, který ukazuje na jiné pravidlo. Tento cyklus dovoluje NAPTR záznamům přírůstkově specifikovat kompletní pravidlo. Tato přírůstková pravidla mohou být postoupena a to dovoluje ostatním entitám specifikovat pravidla. Toto znamená, že jedna entita nepotřebuje rozumět všem pravidlům.

Například budu si chtít zpravovat NAPTR záznamy sám ve své doméně, proto požádám svého VoIP operátora, aby na moje telefonní číslo vytvořil NAPTR záznam, který nebude mít žádný příznak v poli „flags“ (bude rekurzivní) a nebude obsahovat žádný regulární výraz v poli „regexp“. V poli „replacement“ bude uvedena moje doména a na této doméně si já vytvořím potřebné záznamy. Díky rekurzivnosti výběru NAPTR záznamu si mohu měnit potřebné záznamy ve vlastní doméně bez toho, abych musel při změně kontaktovat svého operátora.

Algoritmus začíná se zadaným řetězcem a známým klíčem (zde doménovým názvem). Záznamy pro tento klíč jsou přijaty a hned jsou vyřazeny ty záznamy, kde jsou neznámé služby

(pole services) a neznámé příznaky (pole flags). Zbývající záznamy jsou seřazeny podle pořadí (pole order). Dále mohou být záznamy se stejnou hodnotou pořadí seřazeny podle preferencí (pole preference).

Záznamy jsou porovnávány v seřazeném pořadí, dokud není nalezena shoda. Shoda nastane ve dvou případech a to:

1. záznam má vyplněné pole replacement na místo pole regexp
2. regulární výraz z pole regexp odpovídá řetězci od klienta

Musí se použít první nalezená shoda. Pokud je nalezená shoda, pak se prozkoumávají služby zdali obsahují žádoucí výsledek. Pokud ano, tak je pravidlo použito na cílový řetězec, pokud ne, pak se proces zastaví. Doména, která je výsledkem regulárního výrazu je použita jako doména dalšího cyklu přes tento algoritmus.

Tento cyklus je velmi důležitý, protože komplexní pravidla jsou tvořena jednotlivými záznamy při průchodu cyklu. Až příznaky (pole flags) jednoduše rozhodnou ve kterém bodě cyklus skončí. Protože příznaky jsou platné na kterékoliv úrovni, tak se ve většině případů používají hned na první úrovni a necyklí se. Toto je vhodné ve většině případů.

### 4.4.3 NAPTR záznamy s ENUM

Výsledkem záznamu s ENUM musí být adresa služby, proto se s NAPTR záznamem používá příznak U. Pole services v záznamu NAPTR je složeno z řetězce E2U (e164 to URI) a z názvu služby systému ENUM. Služby systému ENUM, které lze uvádět v NAPTR záznamech, jsou definovány <http://www.iana.org/assignments/enum-services>. Některé služby systému ENUM jsou uvedeny v následující tabulce.

Název	Typ	Podtyp	URI	Popis
sip	sip	není	sip:	SIP kontakt
h323	h323	není	h323:	H323 kontakt
email	email	mailto	mailto:	Email kontakt
sms	sms	mailto tel	mailto: tel:	SMS kontakt, který se může poslat buď jako mail, nebo jak sms na telefon.

*Tabulka 5: Služby systému ENUM*

### Praktická ukázka v konzoli

V příkazovém řádku pomocí programu `host` získáme URI adresu čísla +420541141108.

```
$ host -t NAPTR 8.0.1.1.4.1.1.4.5.0.2.4.e164.arpa
```

```
8.0.1.1.4.1.1.4.5.0.2.4.e164.arpa has NAPTR
```

```
record 100 50 "u" "E2U+sip" "!^\\+(.*)$!sip:\\1@vutbr.cz!" .
```

```
8.0.1.1.4.1.1.4.5.0.2.4.e164.arpa has NAPTR
```

```
record 200 50 "u" "E2U+h323" "!^\\+(.*)$!h323:\\1@h323.vutbr.cz!" .
```

Z uvedeného výsledku je patrné, že telefonní číslo má dva záznamy. První má větší prioritu 100 a druhý nižší prioritu 200. Hodnota pole `order` není zde důležitá, protože zde nejsou dva záznamy se stejnou prioritou. Pole `flags` má příznak `U`, což znamená, že výsledkem bude URI adresa. Následuje pole `service`, kde v prvním případě hodnota `E2U+sip` znamená adresu SIP a ve druhém případě `E2U+h323` znamená adresu `h323`. Potom následuje pole `regexp`, kde je oddělovačem znak „!““. Regulární výraz je „`^\\+(.*)$`“, který označuje libovolnou směs znaků a na začátku „+“, je aplikován na řetězec klienta (telefonní číslo +420541141108). Na tuto směs znaků bez úvodního plus je nastavena zpětná reference s číslem 1. Tato zpětná reference je následně použita v nahrazujícím řetězci, kde `\\1` je nahrazeno jejím obsahem. Výsledná adresa URI je v případě prvního záznamu `sip:420541141108@vutbr.cz` a ve druhém případě je to `h323:420541141108@vutbr.cz`.

## 4.5 DNS SRV záznamy se SIP

Nyní máme pomocí systému ENUM k dispozici adresu SIP, ale ještě nevíme, jak ji používá klient nebo proxy server. Díky SRV záznamu můžeme v DNS systému specifikovat informaci o dostupných službách jako je SIP nebo XMPP. Tyto protokoly vyžadují od klientů podporu pro SRV záznamy.

### Formát záznamu

`_sluzba._protokol.nazev ttl třída SRV priorita váha port cíl`

kde jednotlivé názvy mají následující význam:

- `sluzba` – symbolické jméno pro službu
- `protokol` – protokol požadované služby, typicky `tcp` nebo `udp`
- `nazev` – doménové jméno pro které je záznam platný

- ttl – standardní DNS pole ttl
- třída – standardní DNS pole, zde vždy IN
- priorita – priorita, při více záznamech, menší hodnota znamená větší priorita
- váha – při stejné prioritě relativní váha
- port – tcp nebo udp port, na kterém je služba dostupná
- cíl – doménové jméno serveru, na kterém je služba dostupná

Pro doménu z adresy SIP sip:420541141108@vutbr.cz si můžeme SRV záznam ukázat v konzoli zadáme-li následující příkaz:

```
$ host -t SRV _sip._udp.vutbr.cz
_sip._udp.vutbr.cz has SRV record 0 1 5060 sip.vutbr.cz.
```

Tento příklad ukazuje, že pokud klient provede dotaz do systému DNS na SRV záznam pro doménu \_sip.\_udp.vutbr.cz, tak dostane odpověď, že služba SIP je dostupná na portu 5060 na serveru sip.vutbr.cz

## 4.6 Nastavení Asterisku na překlad ENUM

Pro nastavení podpory ENUM musíme v Asterisku udělat několik věcí. Asterisk má v sobě implementovanou funkci „EnumLookup“, která vrací ENUM záznamy. Tuto funkci lze využít v číslovacím plánu. Vzhledem k tomu, že funkce vrací vždy jen jeden záznam, tak musíme vytvořit makro, kterému předáme v parametru telefonní číslo a makro bude postupně zkoušet všechny ENUM záznamy vytáčet. Pokud se to nepovede vytočíme číslo přes náš standardní účet VoIP operátora. Dále musíme povolit DNS překlad srv záznamů.

Funkce EnumLookup(číslo,typ,pořadí), kde:

- číslo – telefonní číslo včetně národního formátu a plus na začátku (pro ČR +420)
- typ – je typ služby, např. sip, iax2, h323 nebo all pro všechny typy
- pořadí – zde zadáme pořadí záznamu, který chceme, pokud zde uvedeme písmeno c, pak funkce vrátí počet záznamů pro dané číslo a typ služby

V souboru sip.conf v sekci [general] nastavíme volbu:

```
srvlookup=yes
```

Dále si vytvoříme makro v souboru extension.conf:

[makro-enum]

```
exten => s,1,Set(sipcount=${ENUMLOOKUP(+${ARG1},sip,c)}|counter=0)
```

```
exten => s,n,While("${counter}"<"${sipcount}")
```

```
exten => s,n,Set(counter=${counter}+1)
```

```
exten => s,n,Dial(SIP/${ENUMLOOKUP(+${ARG1},sip,${counter}))
```

```
exten => s,n,EndWhile
```

`${ARG1}` je první parametr makra enum – telefonní číslo bez úvodního plus a s mezinárodním předčíslem. V prvním kroku nastavíme pomocí funkce Set proměnnou sipcount na počet ENUM záznamů s telefonním číslem v `${ARG1}` a službou sip. A ještě si nastavíme pomocnou proměnnou counter na nulu. Dále už jen cyklíme, vždy zvyšujeme counter o jedničku a začneme vytáčet. Teoreticky tento cyklus pokračuje i pokud jsme se úspěšně dovolali a existuje další záznam, jenže na konci hovoru oba klienti ukončí hovor, takže Asterisk ukončí tuto smyčku.

Dále vytvoříme v extensions.conf volbu, aby se ENUN používal vždy při vytočení českého nebo mezinárodního čísla. Volbu vložíme do výchozího kontextu. Volba je následující:

```
exten => _XXXXXXXXXX,1,Macro(enum,420${EXTEN})
```

```
exten => _+X.,1,Macro(enum,${EXTEN:1})
```

První řádek je volání 9 ciferného čísla, kde předpokládáme, že je to české číslo, proto doplníme českou mezinárodní předvolbu a voláme Makro enum.

Druhý řádek je volání libovolného čísla s úvodním plus, zde předpokládáme, že jde o číslo v mezinárodním formátu. Při volání makra odděláme plus, protože v makru se přidává.

## 5 Testování

Testování a ověřování funkčnosti jsem prováděl v laboratoři FIT a také doma, kde se objevil jako největší problém moje připojení k internetu. Protože mám od poskytovatele neveřejnou IP adresu a on ji NATuje 1:1 na veřejnou. V tomto případě bylo zapotřebí nastavit Asterisk tak, aby ve zprávách SIP uváděl veřejnou IP adresu. Poté se již bylo možné dovolat, ale byl problém s hlasem. Zde bylo zapotřebí nastavit, aby Asterisk posílal přenos hlasu přes proxy server a ne přímo.

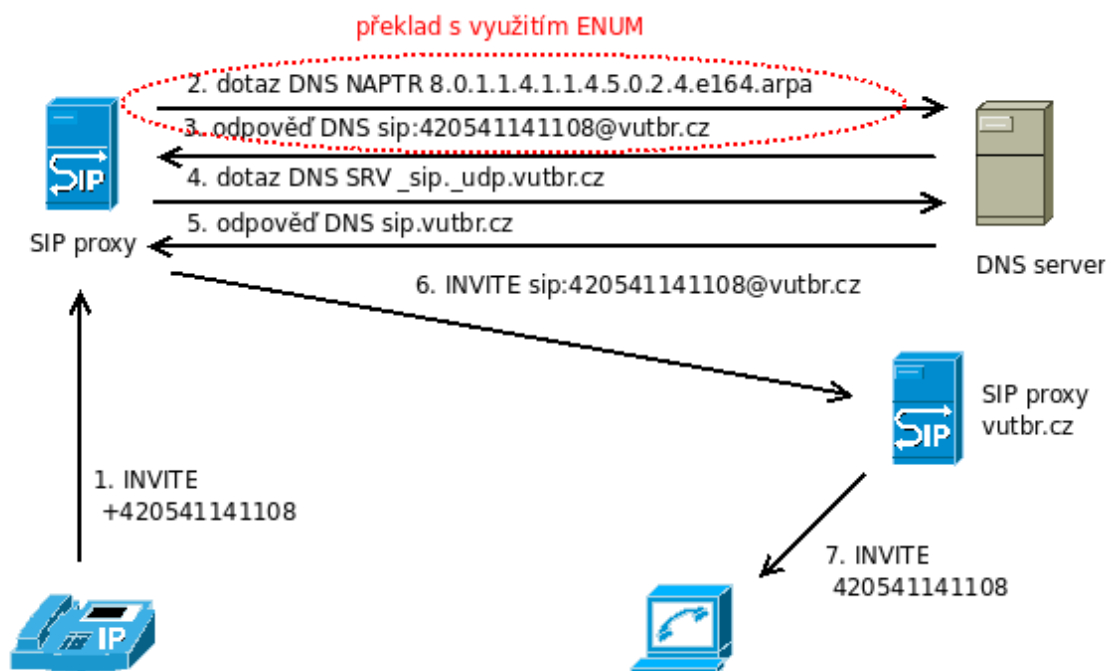
### 5.1 Testování ENUM

Testování ENUM jsem prováděl doma pomocí serveru Asterisk s překladem ENUM. Přímo sdružení „cz.nic“ má na svých stránkách uvedeno, že má několik telefonních čísel, na které se lze dovolat pomocí ENUM. Pokud se na tyto čísla dovoláme pomocí ENUM, tak uslyšíme slovo ENUM. Pokud se dovoláme přes PSTN, uslyšíme slovo PSTN. Telefonní čísla jsou následující:

- 222745120 – poslech hudby (sip:hudba@nic.cz)
- 222745121 – echo test (sip:echo@nic.cz)
- 222745122 – systém IVR a tím lze otestovat DTFM (sip:ivr@nic.cz)

Bohužel jsem při mém testování narazil na několik problémů. První z nich, který mi zabral nejvíce času, byl sice triviální, ale nikde jsem se o něm nedozvěděl. Jde o to, že pokud nemáte uvedeno v hlavičce SIP v poli From české telefonní číslo, tak se nespojíte. Pravděpodobně je to ochrana proti hlasovému spamu, ovšem na druhou stranu, když jsem uvedl české číslo, které nebylo moje, tak jsem se dovolal. A to i přesto, že jsem nastavil v adrese SIP IP adresa localhostu (127.0.0.1). Další problém spočíval v tom, že Asterisk měl povolen překlad SRV záznamů (kvůli ENUM překladu), jenže můj VoIP operátor (802.cz) je nemá vytvořené, a tak jsem se Asterisk nemohl přihlásit na můj účet. Další volání do akademické sítě CESNET probíhalo bez problémů.

Na následujícím obrázku je ukázáno, jak testování probíhalo.



Obrázek 13: Vytočení čísla s ENUM překladem

### 5.1.1 Ukázka SIP komunikace se zamítnutím

V této ukázce je jasně vidět ze SIP server odmítá požadavky na spojení, když není v položce from uvedeno české číslo a hlásí chybu 407 proxy authentication required.

```
Reliably Transmitting (no NAT) to 217.31.204.193:5060:
```

```
INVITE sip:hudba@nic.cz SIP/2.0
```

```
Via: SIP/2.0/UDP 127.0.1.1:5060;branch=z9hG4bK7d4385bb;rport
```

```
From: "100" <sip:100@127.0.1.1>;tag=as790125e8
```

```
To: <sip:hudba@nic.cz>
```

```
Contact: <sip:100@127.0.1.1>
```

```
Call-ID: 375e36605f3938a42f955ca031dd2323@127.0.1.1
```

```
CSeq: 102 INVITE
```

```
<-- SIP read from 217.31.204.193:5060:
```

```
SIP/2.0 407 Proxy Authentication Required
```

```
Via: SIP/2.0/UDP
```

```
127.0.1.1:5060;branch=z9hG4bK7d4385bb;received=217.112.160.214;rport=5060
```

```
From: "100" <sip:100@127.0.1.1>;tag=as790125e8
```

```
To: <sip:hudba@nic.cz>;tag=as0788a616
Call-ID: 375e36605f3938a42f955ca031dd2323@127.0.1.1
CSeq: 102 INVITE
```

## 5.1.2 Ukázka SIP komunikace s telefonním číslem

V této ukázce je vidět, že pokud zadáme do SIP klienta české telefonní číslo, tak se dovoláme bez problému pomocí ENUM na testovací čísla sdružení „cz.nic“.

```
Reliably Transmitting (no NAT) to 217.31.204.193:5060:
INVITE sip:hudba@nic.cz SIP/2.0
Via: SIP/2.0/UDP 127.0.1.1:5060;branch=z9hG4bK7984bfde;rport
From: "571117395" <sip:571117395@127.0.1.1>;tag=as07dfed63
To: <sip:hudba@nic.cz>
Contact: <sip:571117395@127.0.1.1>
Call-ID: 1d963843407b4373190ef1a37f092496@127.0.1.1
CSeq: 102 INVITE
```

```
<-- SIP read from 217.31.204.193:5060:
SIP/2.0 100 Trying
Via: SIP/2.0/UDP
127.0.1.1:5060;branch=z9hG4bK7984bfde;received=217.112.160.214;rport=5060
From: "571117395" <sip:571117395@127.0.1.1>;tag=as07dfed63
To: <sip:hudba@nic.cz>
Call-ID: 1d963843407b4373190ef1a37f092496@127.0.1.1
CSeq: 102 INVITE
```

```
<-- SIP read from 217.31.204.193:5060:
SIP/2.0 200 OK
Via: SIP/2.0/UDP
127.0.1.1:5060;branch=z9hG4bK7984bfde;received=217.112.160.214;rport=5060
From: "571117395" <sip:571117395@127.0.1.1>;tag=as07dfed63
To: <sip:hudba@nic.cz>;tag=as465cee0f
Call-ID: 1d963843407b4373190ef1a37f092496@127.0.1.1
CSeq: 102 INVITE
```

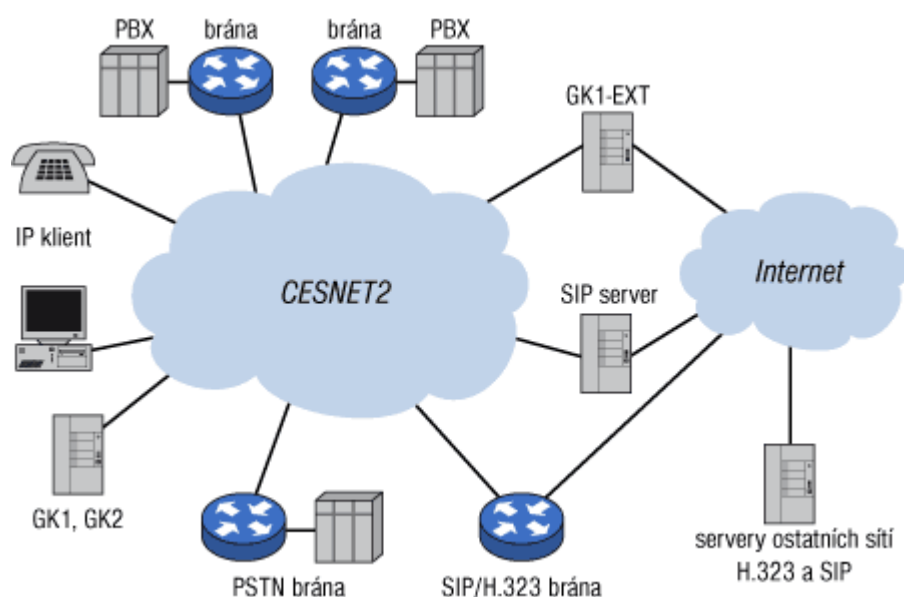


## 6 Akademická síť CESNET

V této kapitole ukážu možnosti propojení serveru Asterisk s překladem ENUM do sítě CESNET. Akademická síť CESNET je národní vysokorychlostní síť. Do této sítě jsou připojeny vysoké školy, Akademie věd České republiky a také i některé střední školy, nemocnice a knihovny. Odtud je patrné, že tato síť slouží především pro vědu, výzkum, vývoj a vzdělávání. Tato síť propojuje největší města v České republice rychlostí až 10 Gb/s a nabízí tím svým uživatelům vysoký komfort ve spojení v rámci sítě.

### 6.1 Struktura VoIP v síti CESNET

VoIP síť se začala budovat v roce 1999 v síti CESNET. V současnosti vybudovaná infrastruktura obsahuje 2 důležité H.323 prvky. Jeden slouží pro připojení do GDS (Global Dialling Scheme), což je nejrozšířenější systém v Evropě pro komunikaci mezi univerzitami. Druhý H.323 prvek slouží pro video konference a slouží také jako vzájemné propojení mezi některými univerzitami a CESNET sítí.



Obrázek 14: Struktura VoIP sítě CESNET

Dále je součástí SIP infrastruktura, která má multidoménový SIP server. Tento server obsluhuje domény cesnet.cz, fel.cvut.cz, ics.muni.cz, czu.cz a další. Mimo obsluhu těchto domén, také směruje hovory do SIP sítí různých VoIP operátorů, funguje tedy jako SIP brána.

Vzhledem k tomu, že jsou zde využívány oba protokoly je nutný překladač. Tento překladač je nasazen v podobě serveru Asterisk. Oblast překladu je pro CESNET velmi důležitá, protože plánují přechod na SIP.

Význam zkratk v obrázku č. 14:

- GK1,GK2 – jsou interní centrální H.323 řídicí prvky (platforma cisco), zde jsou připojeny hlasové brány
- GK1-EXT – je hraniční H.323 řídicí prvek (platforma pc s OS Linux a nainstalovaným GnuGK), tento server obsluhuje IP klienty a dále je spojen s okolním světem přes GDS
- SIP server – je multidoménový proxy server (platforma ps s OS Linux a nainstalovaným SIP express Router), tento server propojuje síť CESNET s komerčními VoIP operátory a dalšími institucemi přes internet
- Brány – jsou to brány institucí (platforma cisco), umožňují duální příjem hovorů (SIP i H.323), protože starší verze IOS SIP nedostatečně podporují, tak musí některé instituce přijímat hovory v H.323 a tudíž zatěžují překladové brány
- SIP/H.323 brány – běží na platformách cisco nebo pc s Linuxem a nainstalovaným Asteriskem.
- IP klienti – různá směs softwarových i hardwarových klientů

## 6.2 Podpora ENUM

V roce 2007 je podpora ENUM v síti CESNET v testovacím provozu. Převod domén na oprávněné držitele je dle výroční zprávy za rok 2007 téměř hotový. Směrování na SIP prvcích do domén e164.arpa je podporováno. CESNET se účastnil konference o službě ENUM, pořádaného sdružením „cz.nic“. CESNET také podporuje myšlenku o možné náhradě stávajícího systému GDS systémem ENUM.

## 6.3 Možnosti propojení

V souvislosti s možnostmi propojení systému Asterisk s ENUM překladem, na který se mohou přihlašovat akademičtí pracovníci FIT, se síti CESNET, je v podstatě zbytečný. Protože většina institucí má vlastní SIP brány, na které jsou směrovány ENUM záznamy. Daleko vhodnější se jeví pouze tento server umístit na páteřní síť FIT, kde bude možnost ho napojit na nějakou adresářovou službu (např. LDAP).

## 7 Závěr

Cílem této práce bylo nakonfigurovat SIP server Asterisk, který bude komunikovat s VoIP sítí FIT. A dále, aby se na tento server mohli přihlašovat zaměstnanci FIT, kteří potom budou moci telefonovat na VUT zdarma.

Začal jsem seznamováním se s protokolem SIP, pak následovala konfigurace serveru Asterisk pro připojení jednotlivých klientů. Zde nastal problém při komunikaci mezi klienty v internetu a klienty ve vnitřní síti, ke které byl server také připojen. Problém vznikl, protože jsem danou konfiguraci testoval na internetovém připojení s NATem 1:1 a zprávy SIP obsahují IP adresu, která se nemění při průchodu přes NAT. Problém byl vyřešen vhodnou konfigurací NATu na serveru Asterisk.

Další část práce se týkala překladu telefonních čísel na identifikátor URI pomocí systému ENUM. Zde jsem se musel seznámit s DNS záznamy NAPTR, které se používají v systému ENUM. Velmi zajímavá část byla o algoritmu výběru záznamu NAPTR. Při testování jsem narazil na problém s voláním na zkušební telefonní čísla ENUM, které provozuje sdružení „cz.nic“. Problém byl v tom, že při pokusu o sestavení spojení server zamítl požadavek. Po určitém čase testování jsem přišel na to, že je potřeba nastavit české telefonní číslo v poli „From“ v hlavičce SIP a pak už spojení šlo sestavit.

Současný stav obsahuje funkční konfiguraci, která umožňuje přihlašování uživatelů a dokáže převádět telefonní čísla pomocí systému ENUM. Další vývoj projektu by mohl obsahovat napojení na školní databázi uživatelů, která je v adresáři LDAP.

# Literatura

- [1] J. Rosenberg, et al.: SIP: Session Initiation Protocol, RFC 3261, 2002.
- [2] A. Gulbrandsen, et al.: A DNS RR for specifying the location of services (DNS SRV), RFC 2782, 2000
- [3] M. Mealling, R. Daniel: The Naming Authority Pointer (NAPTR) DNS Resource Record, RFC 2915, 2000
- [4] Dokumentace IP telefonie dostupná na <https://sip.cesnet.cz> 20.7.2008
- [5] Asterisk dokumentace dostupná na <http://www.voip-info.org/wiki-Asterisk> 20.7.2008
- [6] Asterisk dokumentace dostupná na <http://www.asterisk.org/> 20.7.2008
- [7] J. Gruntorád: Roční zpráva o řešení výzkumného záměru, dostupná na <http://www.cesnet.cz/doc/2007/zprava/vps.html> 20.7.2008
- [8] M. Vozňák: Signalizace SIP, dostupný na [http://www.phonet.cz/archiv/dok\\_osta/ipt-2006\\_Signalizace\\_SIP.pdf](http://www.phonet.cz/archiv/dok_osta/ipt-2006_Signalizace_SIP.pdf) 20.7.2008
- [9] J. V. Meggelen, L. Madsen, J. Smith: Asterisk The future of Telephony, O'Reilly 2007
- [10] R. Kaza, S. Asadullah: Cisco IP Telephony, Cisco Press 2005.
- [11] H323 dokumentace dostupná na <http://en.wikipedia.org/wiki/H.323> 20.7.2008
- [12] OpenH323 projekt <http://sourceforge.net/projects/openh323>
- [13] J. Rosenberg, H. Schulzrinne: An Extension to the Session Initiation Protocol (SIP) for Symmetric Response Routing, RFC 3581, 2003
- [14] Dokument: Jak si vytvořit účet: dostupný na <https://sip.cesnet.cz/cs/cesnet/serreg> 20.7.2008
- [15] P. Faltstrom, M. Mealling: The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM), RFC 3761, 2004
- [16] IEEE, "IEEE Standard for Information Technology – Portable Operating System Interface (POSIX) - Part 2: Shell and Utilities (Vol. 1)", IEEE Std 1003.2-1992, January 1993.

# Seznam příloh

Příloha 1: Instalace Asterisku

Příloha 2: Ukázková konfigurace

Příloha 3: Dokumentace Asterisku

## Příloha 1: Instalace Asterisku

Použil jsem linuxovou distribuci Debian 4.0, kde je asterisk ve verzi 1.2.13, kde se provede instalace použitím příkazů:

```
#apt-get install asterisk
```

Po konfiguraci se musí nastavit, že je asterisk nakonfigurován. V souboru `/etc/default/asterisk` nastavíme `RUNASTERISK=yes`.

Nyní můžeme spustit asterisk:

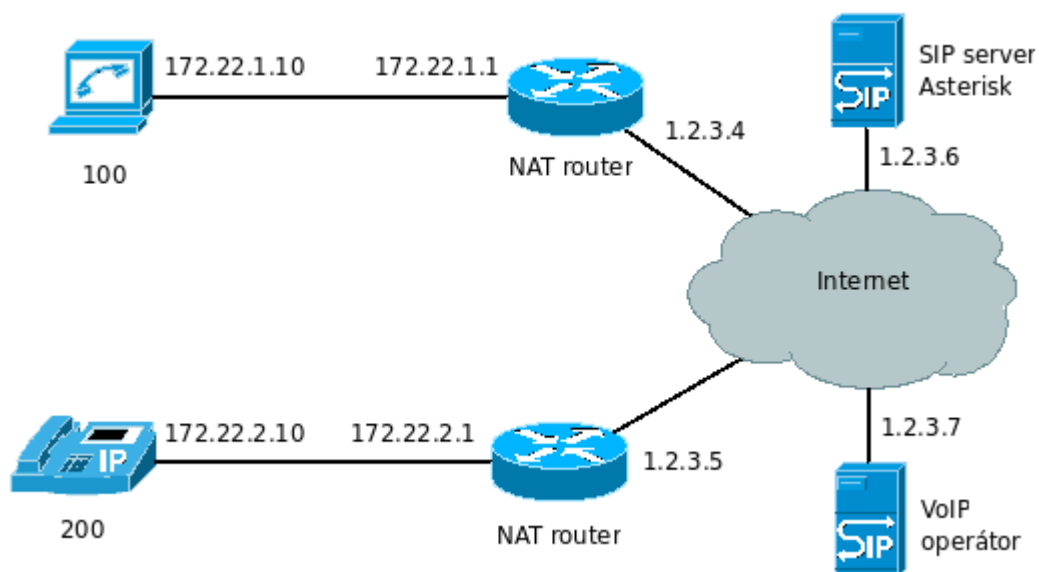
```
#/etc/init.d/asterisk start
```

Při změně konfigurace provedeme restart asterisku:

```
#/etc/init.d/asterisk restart
```

## Příloha 2: Ukázková konfigurace

Následující konfigurace ukazuje připojení Asterisku k VoIP operátorovi, který nám přidělil číslo. Můžeme tedy komunikovat oběma směry. Tzn. zachovají se dvě čísla 100 a 200 pro dva klienty, vše ostatní se bude volat přes našeho VoIP operátora. A příchozí hovor na VoIP operátora se přeměruje na klienta s číslem 100. Dále budeme předpokládat, že oba klienti jsou za NATem oproti našemu serveru Asterisk.



soubor extensions.conf

[hostname]

exten => 100,1,Dial(SIP/100) ; na lince 100 vytočení uživatele se jménem 100

exten => 200,1,Dial(SIP/200)

exten => \_X.,1,Dial(SIP/\${EXTEN:0}@xxxxxxxx) ; je popsáno v předešlé kapitole

exten => \_X.,n,Hangup

[incoming] ; Definuje kontext pro příchozí hovory. To, že tento kontext je pro příchozí hovory

; není dáno jeho jménem ale tím, že je uveden v sip.conf ve volbě register viz níže.

exten => xxxxxxxx,1,Dial(SIP/100)

exten => xxxxxxxx,n,Hangup

soubor sip.conf:

[general]

context=hostname ; Zde se nastavuje název serveru (hostname) nebo doménové jméno.  
bindport=5060 ; UDP port, na kterém bude server naslouchat.  
bindaddr=0.0.0.0 ; IP adresa, na které bude server naslouchat  
domain=172.22.1.1 ; tzv. Lokální domény.  
register => xxxxxxxx:heslo@voip.example.com:5060/**incoming**  
 ; jméno:heslo@poskytovatel:port/kontext

; Konfigurace klientů.

[100] ; Identifikace klienta (použije se jako přihlašovací jméno)  
type=friend ; Typ klienta  
secret=passwd100 ; Heslo klienta.  
host=dynamic ; IP adresa odkud se klient přihlašuje, nebo dynamic.  
canreinvite=no ; hovorový kanál jde přes Asterisk  
qualify=no ; zabraňuje vypršení otevřeného spojení na NAT serveru  
nat=yes ; předpokládá NAT

[200]

type=friend  
secret=passwd200  
host=dynamic  
canreinvite=no  
qualify=no  
nat=yes

[xxxxxxxx] ; uživatelské jméno pro VoIP operátora

type=friend

username=xxxxxxxx ; uživatelské jméno pro VoIP operátora

fromuser=xxxxxxxx ; uživatelské jméno pro VoIP operátora

fromdomain=voip.example.com ; VoIP doména operátora

host=voip.example.com ; VoIP server operátora

secret=heslo

insecure=very ; tato volba způsobí, že u příchozích hovorů nebude vyžadována autorizace

nat=no

qualify=no

context=incoming ; nastaví kontext, musí být stejný jako u volby registr v sekci general!



## Příloha 3: Dokumentace Asterisku

Za vývojem softwaru Asterisk je firma Digium. Tato firma se živí podporou tohoto softwaru. Dokumentace sice existuje, ale v mnoha případech je neúplná a nejsou v ní popsány všechny volby. Existují i komunitní servery s popisem voleb, ale bývá tam často napsáno, že tyto funkce nejsou zdokumentovány.

### Další volby souboru sip.conf ze sekce [general]

Název volby	Popis volby
realm = hostname	Doména, která se zobrazuje uživatelům.
language = en	Výchozí jazyk, který se používá ve funkcích Playback()/Background(). Tyto funkce se používají v číslovacím plánu
- maxexpire = 60	Maximální doba (v sekundách) vypršení registrace klienta.
- outboundproxy = 1.2.3.4	IP adresa nebo doménové jméno odchozího SIP proxy serveru.
outboundproxyport = 8000	Port odchozího SIP proxy serveru.
pedantic = yes no	Zapíná podrobnou kontrolu id hovoru, SIP hlaviček, atd..
srvlookup = yes no	Zapíná možnost překladu DNS SRV záznamů. Např. když budeme volat na sip:user@example.com, tak SRV záznam může odkazovat na sip.example2.com. Momentálně pokud zapneme tuto volbu, tak Asterisk přečte první záznam bez ohledu na priority.
- tos = lowdelay   throughput   reliability   mincost   none	(Type of Service) Nastavuje IP QoS parametr pro odchozí hlasové pakety
- musicclass = music	Třída hudby specifikovaná v musiconhold.conf.

### Další volby souboru sip.conf z klientské sekce

Název volby	Popis volby
accountcode = xxx	Identifikační řetězec uživatele, který se ukládá při účtování hovorů.
allow = G711a	Seznam všech povolených kodeků. Pokud chceme povolit jen některé, musíme nejdříve vše zakázat (disallow=all).
disallow = gsm	Seznam všech zakázaných kodeků.

auth = xxx	Řetězec, který se posílá v SIP zprávě při autentizaci. Např.: <user>:<secret>@<realm>
callerid = Jan Novak	Identifikace volajícího.
call-limit = 10	Počet současných hovorů.
dtmfmode = inband   info   rfc2833	Znamená mód posílání stisknutých kláves během hovoru. Využívá se například pro hlasovou schránku nebo IVR. <ul style="list-style-type: none"> <li>● inband: Posílá audio signál stisknutých kláves, lze použít pouze u kodeků G711 (bez komprese). Protože v jiném kodeku může dojít ke ztrátě vlivem komprese.</li> <li>● info: Používá sip info zprávy.</li> <li>● rfc2833: Používá rtp kanál pro posílání dtmf signálů.</li> </ul>
incominglimit = 2	Počet současných příchozích aktivních hovorů pro SIP klienta.
outgoinglimit = 2	Počet současných odchozích aktivních hovorů pro SIP klienta.
language = en	Kód jazyka definovaný v indications.conf. Nastavuje jazyk pro hlasové výzvy.
mailbox = voicemail@context	Nastaví kontrolu hlasové schránky. Pokud se v ní nacházejí zprávy pošle zprávu SIP simple klientovy, který ji musí podporovat.
md5secret = hash	Nastavuje heslo v MD5 hash formě. Hash je složen následovně <user>:<realm>:<secret>. Pokud není realm zadáno použije se Asterisk.
port = 5060	SIP port klienta. Využívá se, pokud se klient neuauthORIZUJE, ale má nastavenou pevnou ip adresu (pomoci host) a port. Potom se volá přímo na danou ip a port.
permit = 192.168.1.0/24	Povoluje SIP komunikaci z tohoto rozsahu.
deny = 192.168.1.0/24	Zakazuje SIP komunikaci z tohoto rozsahu.