



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

OPEN SOURCE POBOČKOVÁ ÚSTŘEDNA YATE

OPEN SOURCE PBX YATE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. JAN KOVÁŘ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PAVEL ŠILHAVÝ, Ph.D.

BRNO 2011



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Jan Kovář

ID: 83602

Ročník: 2

Akademický rok: 2010/2011

NÁZEV TÉMATU:

Open source pobočková ústředna YATE

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte možnosti Open Source pobočkové ústředny YATE (Yet Another Telephony Engine). Realizujte její instalaci a konfiguraci na linuxové distribuci Ubuntu včetně analogové karty a karty E1. Na základě získaných zkušeností proveďte podrobné porovnání tohoto open source řešení s PBX Asterisk. Text vlastní práce bude obsahovat podrobné návody instalace a konfigurace dílčích částí. Na základě získaných zkušeností vytvořte zadání dvojice laboratorních úloh, které umožní seznámení s konfigurací a správou ústředny.

DOPORUČENÁ LITERATURA:

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

[2] Bazala, D. Telekomunikace a VoIP telefonie I. Praha: BEN - technická literatura, Praha 2006, ISBN 80-7300-201-9.

[3] YATE - Main - Documentation. [online], [cit. 2010-10-11]. Dostupné z <http://www.yate.null.ro>.

Termín zadání: 7.2.2011

Termín odevzdání: 26.5.2011

Vedoucí práce: Ing. Pavel Šilhavý, Ph.D.

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ANOTACE

Práce pojednává o pobočkové ústředně YATE, popisuje její konfiguraci od prvotní instalace, přes základní nastavení a popis jednotlivých protokolů přes způsob registrace uživatelů, syntaxe pro definování volacího plánu včetně příkladů použití, a demonstraci doplňkových služeb včetně praktického overení funkčnosti v počítačové učebně

Dále jsou vytvořeny dvě vzorové úlohy pro bližší seznámení s principy fungování a nastavení pobočkové ústředny Yate.

Klíčová slova:

pbx, yate, sip, iax, h.323, volací plán, registrace uživatelů

ABSTRACT

The work discusses about the PBX Yate, describes its configuration from the initial installation, through the basic setup and description of the way through the logs of user registration, the syntax for defining the calling plan, including examples, and additional services, including demonstration of practical verification of the functionality in the computer lab.

Are also created two samples job for more familiar with the principles of operation and the Yate PBX settings.

Keywords:

PBX, Yate, SIP, IAX, H.323, calling plan, user registration

BIBLIOGRAFICKÁ CITACE DÍLA

KOVÁŘ, J. Open source pobočková ústředna YATE. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. XY s. Vedoucí diplomové práce Ing. Pavel Šilhavý, Ph.D.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma „Open source pobočková ústředna YATE“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 26.5.2011

.....

podpis autora

Poděkování

Děkuji vedoucímu práce Ing. Pavlovi Šilhavému, PhD., za velmi užitečnou metodickou pomoc a cenné rady při zpracování diplomové práce a především za projevenou ochotu a trpělivost.

V Brně dne 26.5. 2011

.....
podpis autora

Obsah

ÚVOD	9
1. PBX YATE	9
1.1 Instalace balíku YATE	10
1.2 Architektura YATE	11
1.3 Protokol SIP	15
1.4 Protokol IAX	17
1.5 H.323	18
1.6 Směrovací plán	21
1.6.1 Kontexty	23
1.6.2 Příklady použití směrovacích pravidel	23
1.7 Registrace uživatelů:	25
1.8 Podpůrné služby PBX	25
1.8.1 Funkce Call forker	26
1.8.2 Tónový generátor:	26
1.8.3 Záznam a reprodukce audio souborů:	27
1.8.4 Konferenční volání	28
2. Nastavení pro použití HW karet	29
2.1 Konfigurační soubor pro karty Sangoma – wpcard.conf	29
2.2 Konfigurační soubor pro ovladače Zapitel (Dahdi) karet Digium – zapcard.conf	31
2.3 Analogový modul analog.conf	31
2.4 Signalizační konfigurační soubor - ysigchan.con	33
2.4.1 IDSN Trunk	34
2.4.2 Konfigurace SS7	35
3. Srovnání PBX Yate s PBX Asterisk	39
4. Praktická část	41
5. Zadání úlohy 1 – Seznámení s PBX YATE a jeho základní konfigurace	43
6. Zadání úlohy 2 – Volací plán a PBX pomocné funkce	50
Závěr	55
Seznam literatury	56
Seznam použitých zkratk	57
Seznam obrázků	58
Seznam tabulek	59
Seznam příloh	60

ÚVOD

V současné době je použití „opensource“ řešení stále více na vzestupu a výjimkou nejsou ani softwarová řešení pobočkových ústředen ve snaze nahradit klasické pobočkové ústředny, které mají od výroby definované funkce a jejich použití je tak limitované.

Tato práce se zaměřuje na práci s pobočkovou ústřednou YATE, popisuje její konfiguraci od prvotní instalace, přes základní nastavení a popis jednotlivých protokolů. Dále je vysvětlen způsob registrace uživatelů, syntaxe pro definování volacího plánu včetně příkladů použití. Jsou demonstrovány doplňkové služby jako přehrávání a záznam audio souborů či vytváření konferenčních místností.

Praktickou částí této diplomové práce je připojení počítače s nainstalovanou ústřednou Yate k ústředně Asterisk včetně volání mezi ústřednami a popis definování jednotlivých trunků.

Dále jsou vytvořeny dvě vzorové úlohy pro potřeby předmětu Telekomunikační a informační systémy pro bližší seznámení studentů s principy fungování a nastavení pobočkové ústředny Yate.

Veškeré konfigurace jsou prováděny výhradně pomocí konfiguračních souborů.

1. PBX YATE

Yate (Yet another telephony engine) je open source pobočková ústředna nové generace, zaměřená na VoIP a PTSN (Public Switched Telephone Network), mezi jejíž přednosti patří snadná rozšiřitelnost. Podporuje hlasové služby, video, datové služby a také IM (instant messaging). Je napsána v jazyce C++ a podporuje scriptování v různých programovacích jazycích (PHP, Python a Perl knihovny). YATE je šířeno pod licencí GPL [1].

YATE může být použit jako:

- VoIP server
- VoIP klient

- VoIP na PTSN brána (gateway)
- H.323 gatekeeper
- H.323 koncový vícebodový server (endpoint server)
- SIP session border kontroler
- SIP router
- SIP registrační server
- IAX server a klient
- IP Telephony server a klient
- Call center server
- IVR (Interactive Voice Response) engine

YATE umožňuje použití následujících kodeků:

Název kodeku	Parametr pro použití v YATE
G711 μ -law	mulaw
G711 A-law	alaw
GSM	gsm
Microsoft GSM	msgsm
Speex	speex
Lineární Prediktivní Kódování	lpc10
internet Low Bit Rate Kodek	ilbc20 ilbc30
Slin (16b PCM)	slin
G.72X	g723 g726 g728 g729

Tabulka 1: Seznam podporovaných kodeků

1.1 Instalace balíku YATE

Instalace ústředny YATE a její testování probíhala na platformě Linux, distribuce Ubuntu 10.10. Přidání balíčku YATE je možno buď z grafického rozhraní přes Systém\Správa\Správce balíku Synaptic nebo pomocí příkazů v terminálu. Před samotnou instalací je zapotřebí přidání repozitáře – cesty, odkud se má balík YATE nainstalovat – pomocí příkazů:

```
add-apt-repository ppa:vpol/yate
apt-get update
```

Poté je již možno balík YATE nainstalovat, buď již pomocí zmíněného Správce balíků Synaptic (vyhledáním klíčového slova YATE), nebo opět pomocí příkazu terminálu

```
apt-get install yate
```

Po zadání výše zmíněného příkazu nebo po vybrání balíku YATE ve Správci balíků Synaptic se nainstalují následující balíky:

- YATE (ve verzi 3.1.0 0)
- YATE-GT4 (univerzální telefonní klient na bázi YATE)
- LIBYATE3.0.0 (sdílená knihovna pro YATE – ver. 3)

V systému souborů jsou pak pro nás zajímavé soubory umístěny následovně:

- /etc/yate – konfigurační soubory
- /usr/lib/yate – umístění jednotlivých modulů
- /etc/default/yate – soubory s nastavením po spuštění systému

Spouštění:

Ústřednu Yate spustíme z příkazového řádku terminálu pomocí následujícího příkazu a parametrů:

```
yate [options] [commands ...]
```

Výpis všech možných parametrů najdete v příloze X.X. V praxi je pro nás nejdůležitější spouštění s výpisem podrobných, barevně odlišených informací, tedy:

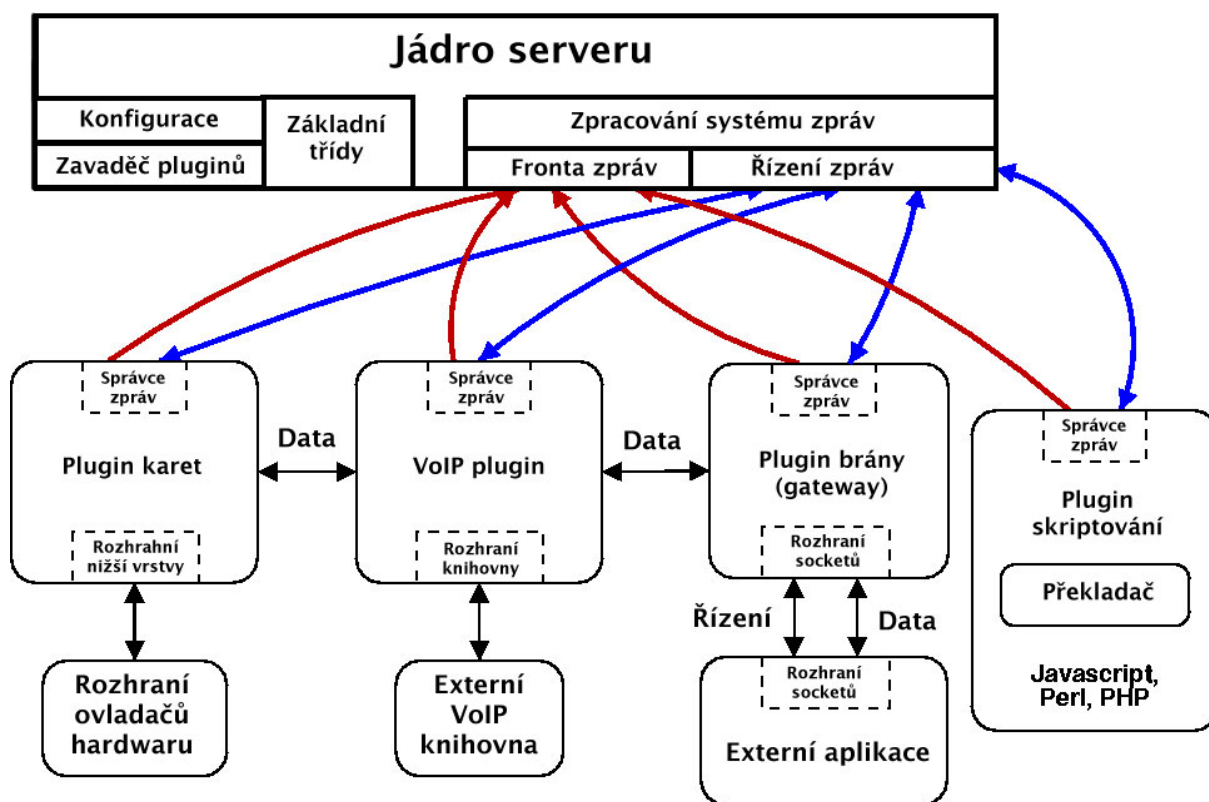
```
yate -vvvvv -Do
```

Přerušování, resp. ukončení běhu YATE provádíme pomocí klávesové zkratky Ctrl + C, restart pak pomocí kombinace kláves Ctrl + \

1.2 Architektura YATE

Nejdůležitějším aspektem Yate je jeho systém zpráv a jejich navázání na moduly. To umožňuje větší flexibilitu, než s prostými funkcemi. Zprávy v Yate mohou mít

libovolný počet parametrů, a mohou být zaslány na více než jeden modul změnou priority.



Obr. 1 – Diagram architektury YATE

(převzato z adresy <http://yate.ro/pmwiki/index.php?n=Main.Architecture>)

Hlavní komponenty můžeme zahrnout do čtyř částí:

- Jádru
- Systém zpráv
- Telefonie
- Moduly Yate

Po spuštění ústředny se načtou moduly ústředny včetně směrovacího plánu. Moduly mohou být načteny všechny nebo je možné nadefinovat, které konkrétní mají být načteny. Nastavení se provádí v konfiguračním souboru *yate.conf*. V sekci *[general]* se načtení všech modulů provede parametrem *modload=enable*. Pokud požadujeme načtení jen určitých modulů, provedeme definici požadovaných modulů v sekci *[modules]* přidáním parametru ve tvaru *nazevmodulu.yate=enable*, např. *ysipchan.conf=enable*. Další možné parametry souboru *yate.conf* jsou následující:

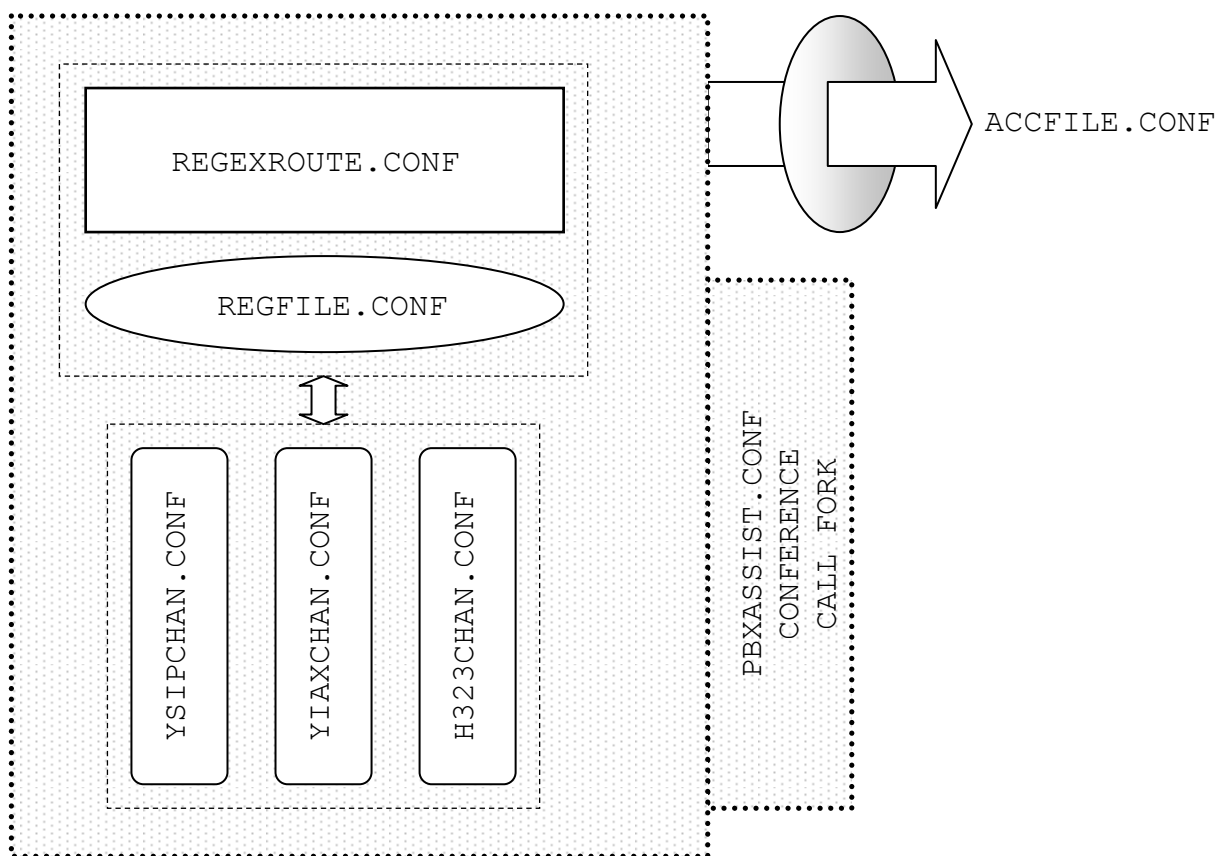
V sekci *[general]* :

- *modpath* – nastavení cesty k modulům, pokud mají jiné než standardní umístění
- *extrapath* – cesta k použití jiných externích modulů
- *restarts* – čas v sekundách, po jehož uplynutí se ústředna restartuje

Sekce *[localsym]* se používá pro „násilné“ načtení modulů i bez definovaných globálních proměnných, opět ve tvaru *nazevmodulu.yate=enable*, sekce *[nounload]* pro moduly, které nemají být uvolněny z paměti. Sekce *[preload]* a *[postload]* se používají pro načtení knihoven před, resp. po načtení modulů Yate a to pomocí parametru */cesta/ke/knihovne/knihovna=enabled*.

V sekci *[debug]* se nastavuje debug level ve tvaru *nazev=level cislo*, např. *sip=level 10*, a v sekci *[telephony]* pak *timeout* pro standardní čas vypršení pro kanály, *maxroute* pro omezení počtu volání, *maxchans* pro omezení počtu kanálů a *dtmfdups* pro povolení nebo zákaz duplikátního DTMF.

Princip z pohledu směrování a použitých protokolů znázorňuje následující obrázek:



Obr. 2 – Blokové schéma provázanosti jednotlivých modulů

Při směrování se nejprve prochází směrovací plán *regexroute.conf*, ve kterém je volací plán definován regulárními výrazy, parametry a funkcemi. Poté se prochází soubor pro registraci uživatelů *regfile.conf*. Ústředna zprostředkovává volání přes protokoly SIP, IAX i H.323. Nastavení jednotlivých protokolů se definuje v souborech *ysipchan.conf*, *yiaxchan.conf* a *h323chan.conf*. Aby bylo možné jednotlivé protokoly použít ve volacím plánu, je zapotřebí vytvoření jejich jednotlivých účtů v souboru *accfile.conf*.

[test_sip]	[test_iax]	[test_h323]
enabled=yes	enabled=yes	enabled=yes
protocol=sip	protocol=iax	protocol=h323

Jednotlivé účty mohou dále mít následující parametry:

username, description, interval, formats, authname, password, number, domain, registrar, outbound a localaddress.

1.3 Protokol SIP

SIP (The session initiation protocol) je signalizační protokol určený k sestavování, modifikování a ukončování multimediálních relací, řadí se mezi protokoly aplikační vrstvy a pracuje na principu klient-server nad transportními protokoly UDP nebo TCP. Protokol je znakově orientovaný, v Yate je zastoupen konfiguračním souborem `ysipchan.conf` a využívá YASS (Yet another SIP stack) knihovnu [1],[2],[3]. Ve volacím plánu je SIP volán následující formou:

```
    cislo=sip/sip:user@ip:port
např: ^2$=sip/sip:diana@10.0.0.1
```

Pro základní zprovoznění SIP protokolu jako registračního serveru je zapotřebí následujícího nastavení:

```
[general] - nastavení globálních proměných
port=5060 - SIP UDP port, na kterém protokol naslouchá
addr=192.168.10.174 - IP adresa, na kterou se pak mohou telefony
                    zaregistrovat
registrar=enable - povolí SIP modulu přijímání registračních žádostí
```

Další možné parametry konfiguračního souboru:

Sekce *[general]*:

```
maxpkt - maximální velikost paketů (524 až 65528)
buffer - velikost buferu
thread - priorita zacházení se SIP vlákny, možné hodnoty lowest, low,
        normal, high, highest
floodevents - počet SIP událostí získaných z řady tzv. trigger flood varovných
              zpráv, výchozí hodnota 20
maxforwards - používá se k zabránění opakovaných volání, výchozí hodnota 20
transfer - povolí zpracování REFER zpráv, typu boolean
options - sestaví a odešle výchozích 200 odpovědí na OPTIONS žádosti, typu
         boolean
```

`prack` – povolení 1xx odpovědí (RFC 3262), typu boolean
`info` – povolení příjmu příchozích INFO zprávy, typu boolean
`fork` – následování prvních 2XX odpovědí na začátku dialogu, typu boolean
`progress` – odeslat "183 Session Progress" po úspěšném směrování, typu boolean
`generate` – povolit libovolné SIP transakce, typu boolean
`nat` – povolit NAT, typu boolean
`ignorevia` – ignorovat Via hlavičky a poslat odpověď zpět ke zdroji, typu boolean
`lazy100` – vytvářet počátečních „100 pokusů“ pro non-INVITE transakce, pokud přenos dorazí před konečnou odpovědí
`dtmfinband` – generovat DTMF na vstupu, typu boolean
`dtmfinfo` – generovat DTMF INFO zprávy k odeslání tónu kláves
`rfc2833` – nabídnout RFC2833 telephone-event jako výchozí
`privacy` – vytvořit privátní SIP záhlaví, typu boolean
`secure` – generovat a přijímat RFC 4568 bezpečnostní deskriptor pro SRTP
`forward_sdp=disable` – zahrnout SDP body pro předávání RTP
`rtp_localip` – IP adresa na lokální RTP, standardně prázdné
`rtp_start` – typu boolean
`multi_ringing` – Přijmout prozatímní (1xx) zprávy i po 180 vyzvánění, typu boolean
`refresh_nosdp` – přijmout relaci obnovení (žádost SDP)

Sekce `[registrar]` řídí chování při použití jako registračního serveru

`expires_min` – čas minimální expirace v sekundách
`expires_def` – výchozí doba platnosti
`expires_max` – max. doba platnosti
`auth_required` – požadována autentizace?, typu boolean
`nat_refresh` – čas v sekundách
`async_process` – asynchronní registrace, typu boolean

Sekce `[sip-t]` pro použití jako SIP-T

`isup` – typu boolean

[codecs] - individuální nastavení kodeků

default - parametr typu boolean, který je možno přiřadit následujícím jednotlivým kodekům pro hromadné povolení nebo zákaz, pokud pro ně nepoužijeme explicitní výraz *enable* nebo *disable*:

mulaw - G711 μ -law (PCMU/8000), alaw - G711 a-law (PCMU/8000),
gsm - GSM 06.10 (GSM/8000), lpc10 - lineární predikční kodek (LPC/8000),
ilbc - Internet Low Bandwidth kodek (iLBC/8000), amr - adaptivní Multi-Rate
3GPP (AMR/8000), slin - Signed Linear 16-bit uncompressed (L16/8000),
g723 - ITU G.723 (G723/8000), g726 - ITU G.726 32-bit (G726-32/8000),
g728 - ITU G.728 (G728/8000), g729 - ITU G.729 all variations (G729/8000),
g729_annexb - G.729 Annex B (VAD)
amr_octet - oktet pro AMR RTP, typu boolean

Sekce [methods] pro zpracování různých SIP metod:

methodname=boolean - název metody, standardní výchozí hodnota *true*

Sekce [hacks] pro práci s poškozenými implementacemi:

ilbc_forced - možné hodnoty *ilbc20* nebo *ilbc30*

ilbc_default - formát pro iLBC pokud je neznámá paketizace

g729_annex - podpora G.729 Annex B při parsování SDP

ignore_missing_ack - ignorovat parametr ACK ve zprávě INVITE, typu boolean

lxx_change_formats=yes

Parametr *port* nastaví port, ke kterému Yate bude vázat signalizaci SIP. Můžeme také zadat IP adresu pro multihomed stroje, v opačném případě bude Yate naslouchat na všech rozhraních.

1.4 Protokol IAX

Protokol IAX byl navržen s ohledem na maximální jednoduchost a snadnost implementace. Jeho cílovou platformou měla být velmi jednoduchá a levná koncová zařízení. Jedná se o binární protokol, který sdružuje signalizaci i hlasová data v jednom kanálu a umožní tak lépe využít šířku pásma a snadno přechází mezi sítěmi

s překladem adres NAT [4], [5]. Protokol je zastoupen konfiguračním souborem *yaxchan.conf*. Při směrování ve volacím plánu má použití následující syntaxi:

```
    cislo=iax/iax:user@ip:port/volané_číslo
např: ^2$=iax/iax:diana@10.0.0.1/123
(při neuvedení portu se použije standardní port 4569)
```

Pro základní funkční konfiguraci je zapotřebí v konfiguračním souboru uvést IP adresu a port, na kterém má ústředna naslouchat příchozí IAX spojení:

V sekci `[general]` pro nastavení globálních proměnných:

`port=4569` – UDP port pro příchozí spojení

`addr=192.168.10.174` – IP adresa serveru

Dále pak mohou být použity následující parametry

`tos=0` – typ služby k nastavení odchozích UDP paketů, hodnota=číslo nebo

`lowdelay, throughput, reliability, mincost`

`read_threads` – počet vláken pro čtení soketů

`event_threads` – počet vláken pro zpracování událostí

`trunk_threads` – počet vláken pro trunkované hlasové pakety

`thread` – priorita pro zacházení s datovými vlákny

V sekci `[formats]` pro individuální nastavení kodeků obdobně jako v předchozí kapitole:

`preferred` – preferovaný formát, dále pak `default, slin, mulaw, alaw, gsm a lpc10`

1.5 H.323

H.323 je součástí doporučení ITU-T H.323, je zastoupen konfiguračním souborem *h323chan.conf* a využívá OpenH323 knihovnu společně s knihovnou PWLib. Umožňuje použití jako gateway, ale také jako gatekeeper. Ve volacím plánu je H.323 voláno následující formou:

```
cislo= h323/user@ip:port
např: ^1$=h323/diana@10.0.0.1:1720
```

Použití jako gateway nebo jako gatekeeper se definuje ve dvou samostatných sekcích, [ep] pro gateway a [kp] pro gatekeeper:

Nastavení pro endpoint:

```
[ep]
```

```
ep = true – aktivace h323 endpoint
```

```
gw = true – deklarovat jako gateway
```

```
port = 1720 – TCP port endpoint
```

```
gkip = 10.10.10.10 – gatekeeper ip adresa
```

```
faststart = true – může způsobit konflikt s povoleným parametrem
```

```
h245insetup
```

```
h245tunneling = true – tunelování H.245
```

Další možné parametry

```
alias
```

```
ident – hostname pro odchozí E.164
```

```
maxconns – počet simultánních spojení (0 = bez limitu).
```

```
h245insetup – může způsobit konflikt s povoleným parametrem faststart,
```

```
typu boolean
```

```
dtmfinband – dekódovat DTMF na vstupu, typu boolean
```

```
silencedetect – detekční algoritmus: none, fixed, adaptive
```

```
gkclient – registrovat na gatekeeper, typu boolean
```

```
password – heslo pro H.235 autentizaci
```

```
gkname – název gatekeeperu pro registraci
```

```
gkttl – gatekeeper time-to-live parametr (v sekundách)
```

```
gkretry – Gatekeeper discovery retry interval (s), 0 = vypnuto
```

Nastavení pro použití jako gatekeeper

```
[gk]
```

```
server = true
```

```
interface1=10.0.0.1  
port = 1719
```

Další možné parametry:

name, ttl, heartbeatdrop a registeredonly

Sekce [general] pro nastavení globálních proměnných

debug, dumpcodecs, vendor, produkt, major, minor, build, status, needmedia, external_rtp – použití externího RTP, tedy modulu *yrtchan.conf*, fallback_rtp – použít nativní RTP pokud externí selže, typu boolean, forward_rtp, maxcleaning,

[codecs] – použití OpenH323 nativního RTP s chybějícími kodeky může způsobit pád ústředny

možné parametry: default, mula, alaw, gsm, msgsm, speex, lpc10, ilbc20, ilbc30, slin, g723, g726, g728, g729

[incoming] – nastavení pro příchozí hovory

context – zařazení do kontextu, maxqueue – maximální délka fronty, called – výchozí číslo, pokud volané číslo není ve volacím plánu

Yate také umožňuje namísto nativního RTP použití externího RTP modulu. Možné nastavení externího RTP modulu *yrtchan.conf* je následující:

V sekci [general] parametry minport a maxport pro určení rozsahu portů, localip, tos – pro určení „type of service“ (lowdelay, throughput, reliability nebo mincost), buffer, autoaddr – pro automatickou změnu odchozích RTP adres, anysrc – pro akceptování jakýchkoliv příchozích SSRC, padding, rtcp, rtcp_interval, drillhole – pro přenos skrze firewall nebo NAT, thread, defsleepp – defaultní „sleep“ čas v ms pro novou relaci, minsleepp – minimální povolený „sleep“ čas

1.6 Směrovací plán

Směrovací plán poskytuje soubor `regxroute.conf`. Stejně jako v PBX Asterisk je možné jednotlivá směrování zahrnout pod určitý kontext. Pokud pro volané číslo kontext nedefinujeme, použijí se výchozí parametry. Základní forma zápisu je následující:

<code>^</code>	začátek řetězce
<code>\$</code>	konec řetězce
<code>.</code>	zastupuje jakýkoliv znak
<code>[list]</code>	jeden znak v daném rozsahu
<code>[^list]</code>	jeden znak nepatřící do daného rozsahu
<code>*</code>	zachytávání předchozího čísla (včetně nuly)
<code>\+</code>	předcházející výraz alespoň jednou
<code>\?</code>	předchozí vyjádření nuly
<code>\{N\}</code>	předcházející výraz přesně N-krát
<code>\{N,\}</code>	předcházející výraz N nebo vícekrát
<code>\{N,M\}</code>	předcházející výraz mezi N a M
<code>\(\)</code>	zachycuje obsažený podvýraz

Příklad zápisu:

```
^00\(.*\)$=iax/\1@internat.ion.al/\0
```

Pro jakékoliv číslo začínající 00 použije protokol IAX. Například pro volané číslo 0099123456 volací plán vrátí

```
iax/99123456@internat.ion.al/0099123456
```

Předchozí zápis demonstruje použití subvýrazů. Zápisem ve volacím plánu je možno vytáčená čísla zahrnout jako subvýraz pomocí syntaxe

```
\( subvýraz \)
```

Ve volacím plánu je potom výraz `\0` nahrazen celým zadaným výrazem, `\1` prvním subvýrazem, `\2` druhým subvýrazem atd., jak je patrné v předchozím příkladě.

V rámci volacího plánu je také možno použít následujících proměnných nebo funkcí:

Proměnné:

- `${address}` – "adresa" (např. analog/pstn/1 ne IP adresa) příchozího hovoru
- `${callsource}` – např. fxs/fxo pro analogové linky
- `${formats}` – kodeky, které příchozí hovor akceptuje
- `${id}` – použitý formát přenosu
- `${ip_host}` – IP adresa
- `${overlapped}`
- `${rtp_forward}` - rtp forwarding
- `${type}` – typ volání (např. záznam)
- `${username}` – jméno uživatele pro příchozí hovor

Příklady funkcí použitelných na „pravé straně“ výrazu:

- `$ ()` vrátí znak ;
- `$ ($)` vrátí znak \$
- `$ (++N)` vrátí N+1
- `$ (--N)` vrátí N-1
- `$ (length , STRING)` vrátí délku řetězce STRING
- `$ (chr , N)` vrátí znak s číselným kódem N
- `$ (eq , VAL1 , VAL2)` vrátí hodnotu "true" pokud VAL1 = VAL2 (číselně), jinak "false"
- `$ (ne , VAL1 , VAL2)` vrátí hodnotu "true" pokud VAL1 != VAL2 (číselně), jinak "false"
- `$ (lt , VAL1 , VAL2)` vrátí hodnotu "true" pokud VAL1 < VAL2, jinak "false"
- `$ (gt , VAL1 , VAL2)` vrátí hodnotu "true" pokud VAL1 > VAL2, jinak "false"
- `$ (transcode , FLAGS , FORMAT1 , FORMAT2 , . . .)` změna formátů na vstupu

Všechny použitelné funkce najdete v příloze 3.

1.6.1 Kontexty

Kontexty ve volacím plánu uvozujeme do hranatých závorek „[“ a “]”. Pro práci s kontexty potom používáme následujících výrazů:

- `if` – podmínka
- `return` – vrátí zpět z aktuálního kontextu bez směrování volání
- `include` – zahrnuje jiný kontext
- `jump` – skok do jiného kontextu (nezahrnuje kontext předchozí jako `include`)
- `match` – upraví odpovídající řetězec na specifikovaný cíl
- `rename` – změna názvu zprávy
- `enqueue` – vloží novou zprávu, parametry vezme ze zprávy předcházející
- `dispatch` – odešle novou zprávu čekající na návrat, parametry jsou převzaty ze staré zprávy a umístěny do nové
- `echo` – zobrazí zadaný řádek ve volacím plánu

1.6.2 Příklady použití směrovacích pravidel

- ▶ `^00\(.*\)$=iax/\1@123.456.789/\0`
Např. pro číslo 0012345 směrování vrátí volání na
`Iax/12345@123.456.789/0012345`
- ▶ `^123456$=return;called=standa`
Pro číslo 123456 dojde k vytočení registrovaného uživatele `standa`
- ▶ `^112$=sig/\0;trunk=link1`
Pro číslo 112 dojde k volání 112 přes E1 trunk `link1`
- ▶ `^[2-9]\(.*\)$=sip/sip:\1@12.56.89;callername = standa`
Výraz `[2-9]` určuje jeden znak, číslici, v rozmezí 2 až 9, např. pro volbu 9345 směrování vrátí
`sip/sip:345@123.456.789`
a změní jméno volaného na `standa`

- ▶ **`^08\(..\) \(.*\) $=iax/\1@10.10.10/\2`**
 Po kombinaci 08 zahrne další 2 znaky pod subvýraz \1 a další znaky pod subvýraz \2, tedy např. pro volání 08123456 vrátí:
`iax/12@10.10.10/3456`

- ▶ **`^0\(.*) $=h323/\1@11.12.13`**
 Všechny hovory začínající na 0 jsou směrovány přes H.323, tedy pro 012345 směrování vrátí:
`h323/12345@11.12.13`

- ▶ **`^\(.\) \(.\) \(.\) $=sip/sip:\0@\3.\2.\1.10`**
 Použití posloupnosti tří zadaných znaků jako IP adresa pro volání, tedy pro 789 směrování vrátí:
`sip/sip:789@9.8.7.10`

- ▶ **`${id}^sip/=if ^123.* $=h323/\0@domena.cz`**
 Pomocí parametru `${id}` je zjištěn protokol volání, pokud odpovídá *SIP*, přejde se na podmínku *if*, která říká, že pokud číslo začíná posloupností 123, je voláno pomocí H.323 s použitím celé zadané posloupnosti.

- ▶ **`.*=echo volající ${caller}, volaný ${called}, ${id}`**
 Výpis parametrů `${caller}`, `${called}` a `${id}` pomocí funkce *echo*

- ▶ **`${rtp_forward}possible=;rtp_forward=true`**
 Zjistí z volání možnost použití *RTP forward*, pokud ano, nastaví parametr *rtp_forward* na hodnotu *true*

- ▶ **`.*=wave/play//home/user/sound/invalid_number.gsm`**
 Pro všechny hovory přehraje zadaný zvukový soubor

1.7 Registrace uživatelů:

Pokud je YATE použit jako registrační server, registrace jednotlivých uživatelů se provádí v souboru `regfile.conf`. YATE nerozlišuje použitý protokol. Zápis pro definici uživatele je následující:

```
[registrovany_uzivatel]
enabled=yes
password=heslo
```

Můžeme dále použít parametry `username` a `formats`. V sekci `[general]` dále nastavujeme:

`autocreate` – automaticky vytvořit vstupy pro registrované uživatele, typu boolean
`auth`, `register`, `route` a `preroute` – nastavení priorit, číselně

Pro volání z vnější strany (např. jiné ústředny) na registrovaného uživatele je zapotřebí ve volací plánu definovat volání uživatele pro určitou volbu. Nelze použít přímý výraz:

```
volane_cislo=registrovany_uzivatel
```

Je zapotřebí použít návratový parametr `return` a nastavení volaného parametrem `called` následovně:

```
volane_cislo=return;called=registrovany_uzivatel
```

Např. pro registrovaného uživatele 101:

```
^101$=return;called=101
```

1.8 Podpůrné služby PBX

Ústředna Yate disponuje modulem PBX Assist, který implementuje funkce PBX. Nastavení se provádí v konfiguračním souboru `pbxassist.conf`. Jsou podporovány operace typu `conference`, `secondcall`, `onhold`, `dialtone`, `transfer` apod.

1.8.1 Funkce Call forker

Tento modul se používá ke směřování k více alternativním cílům, přičemž první volaný účastník, který není obsazen, je spojen. Modul nemá konfigurační soubor a ve volacím plánu se používá následovně:

```
^.*$=fork sip/sip:\0@host1 | lateroute=reg_uzivatel
```

Volání na registrovaného uživatele nelze opět zadat přímo, je zapotřebí použití parametru *lateroute*. V tomto příkladu je použit znak „|“ (roura). Ten způsobí, že je nejprve volána první SIP destinace, pokud ke spojení nedojde (obsazeno, odmítnutí hovoru apod), dojde k volání druhé destinace. Pokud bychom znak roury nepoužili, došlo by k volání na obě destinace zároveň. Funkci fork je možno doplnit o časový parametr, který vynutí změnu destinace po uplynutí dané časové prodlevy:

- |next=N – přepne volání na další destinaci po N milisekundách
- |drop=N – ukončí volání na danou destinaci a přepne volání na další destinaci po N milisekundách

K opuštění modulu `fork` se používá parametru

```
|exec,  
resp. |exec=N
```

```
fork sip/sip:1@host1 |exec=20000 wave/play//queue_full.au
```

1.8.2 Tónový generátor:

YATE umožňuje po vytočení čísla generovat tóny volání (např. vytáčení, obsazeno apod.). Použití ve volacím plánu `regexroute.conf` je standardně následující:

```
tone/parametr
```

příčemž parametry mohou být tyto:

- *dial* – tón vytáčení
- *busy* – tón obsazeno
- *ring* – vyzváněcí tón
- *congestion* – síť obsazena
- *outoforder* – mimo provoz
- *miliwatt* – testovací tón o frekvenci 1kHz
- *silence* - ticho
- *noise* – nízká úroveň bílého šumu

Příklad použití v `regexroute.conf` pro vytáčecí tón při volbě čísla 1:

```
^1$=tone/dial
```

Konfigurační soubor pro tónový generátor `tonegen.conf` v sobě obsahuje přednastavení vyzvánění pro několik různých zemí světa včetně České Republiky. Výběr požadované země se provede zápisem `lang` s daným parametrem, tedy

```
lang=cz
```

1.8.3 Záznam a reprodukce audio souborů:

YATE rovněž umožňuje přehrávání a záznam audio souborů v následujících formátech:

- *.slin* (16-bit signed linear)
- *.alaw* (G.711a)
- *.mulaw* (G.711u)
- *.gsm* (GSM 06.10)
- *.au*
- *.lbc .ilbc20 .ilbc30* (ILBC)

Použití modulu se provádí opět ve volacím plánu pomocí

```
wave/play//cesta_k_souboru
```

pro přehrání audio souboru nebo

```
wave/record//cesta_k_souboru
```

pro záznam.

Tedy např. pro volbu čísla 1 přehraj soubor audiotoplay.slin:

```
^1$=wave/play//path/to/file/audiotoplay.slin
```

1.8.4 Konferenční volání

Jednoduchým způsobem lze také provádět tzv. konferenční volání, tedy spojení několika účastníků zároveň. To se provede volbou určeného čísla z volacího plánu.

Ve volacím plánu se konferenční místnost definuje následovně:

```
conf/místnost
```

tedy např.:

```
^701$ = conf/sales
```

Účastníci volající na číslo 701 jsou spojení do „místnosti“ sales. Pokud by název místnosti nebyl zadán, vygeneruje se náhodný název místnosti ve tvaru

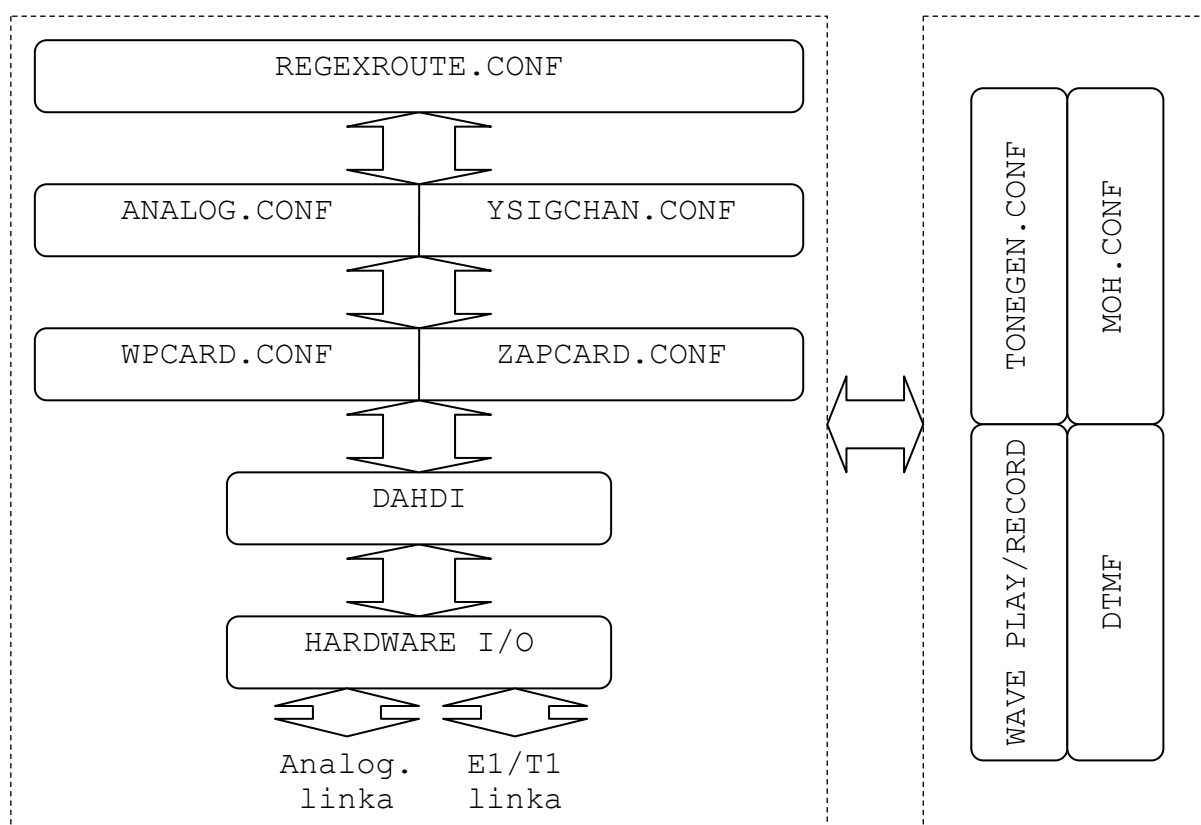
```
x-NUMBER
```

přičemž NUMBER je číslo, které se s každou novou založenou konferenční místností zvyšuje.

2. Nastavení pro použití HW karet

YATE spolupracuje s hardwarovými kartami podobně jako Asterisk, tedy s kartami Sangoma (`wpcard.conf`) a Digium (`zapcard.conf`).

Ty vytvářejí audio kanály (B channel), signalizační kanály (D channel) a případně FXS/FXO analogové okruhy. Kanály se definují v tzv. spanech v jednotlivých konfiguračních souborech pro danou kartu a na ty se pak odkazuje v signalizačním konfiguračním souboru `ysigchan.conf` pro E1/T1 nebo `analog.conf` pro FXS/FXO definováním tzv. linků, resp. trunků, na které poté odkazujeme ve volacím plánu.



Obr. 3 – Znáornění modulů YATE jako funkčních bloků vzhledem k HW kartám

2.1 Konfigurační soubor pro karty Sangoma – `wpcard.conf`

Konfigurační soubor je čten při zpracování žádosti. V parametru `sig` žádosti musí být uveden stávající oddíl v `wpcard.conf` odpovídající s ovladači karty (např. `wanpipe1`). Parametr `siggroup` musí uvádět platné zařízení nakonfigurované v

ovladači (např. w1g1). Obdobná situace nastává i při budování D-kanálů s parametry `voice` a `voicegroup`. YATE nepodporuje analogové kanály FXS/FXO u karet Sangoma [1]. Pro definování spanu je zapotřebí následujícího nastavení:

`[wanpipe1]` – název shodující se s názvem definovaným v ovladači `wanpipe`

`type` - E1 nebo T1

`siggroup` – parametr pro vytvoření D-kanálu definovaným v ovladači `wanpipe`,
např. w1g1

`voicegroup` – parametr pro vytvoření D-kanálu definovaným v ovladači `wanpipe`,
např. w1g2

`voicechans` – rozsah kanálů pro hlas, např. pro E1: 1-15.17-31

`offset` – posun;kladná hodnota přičtena ke každému číslu kanálu

Dále je možno pro daný span definovat parametry:

`readonly` – pouze pro čtení?: `yes/no`

`samples` – počet samplů, které mají být přijaty, slouží k vytvoření vyrovnávací paměti pro čtení dat z rozsahu; celková vyrovnávací paměť bude mít počet bytů daných počtem vzorků, vynásobených počtem hlasových kanálů konfigurovaných pro dané rozpětí; pokud není uvedeno, vypočte se z typu spanu, tzn. pro E1: 50, pro T1: 64

`errormask` – výchozí hodnota 255

`dtmfdetect` – tónový detektor na hlasových kanálech; `yes/no`

V sekci `[general]` můžeme dále definovat následující:

`bitswap` – specifikuje, zda-li odesílaná/přijímaná data jsou po bytu rotována;
možné hodnoty typu `boolean`: `enable/disable`

`idlevalue` – hodnota k vyplnění pokud hlasová data chybí; hodnota v intervalu
0-255, standardně 255

`buflen` – velikost bufferu, standardně 160

2.2 Konfigurační soubor pro ovladače Zapitel (Dahdi) karet Digium – zapcard.conf

U karet Digium YATE podporuje kromě vytváření hlasových a signalizačních kanálů B a D také analogové okruhy FXS/FXO. Stejně jako u konfiguračního souboru pro karty Sangoma, i zde jsou B-kanály a D-kanály sestaveny na požádání (na základě výzev API)[1]. Span musí být definován následujícím způsobem:

```
[span1]
type – typ spanu, E1, T1, FXS nebo FXO
offset – posun, povinný parametr
sigchan – číslo signalizačního kanálu, standardně 16 pro E1 a 24 pro T1
voicechans – rozsah hlasových kanálů, např. (E1: 1-15.17-31, T1: 1-23,
FXO/FXS: 1)
```

Obdobně je možno také nastavit:

```
dtmfdetect – tónový detektor; enable/disable
echotaps – výchozí hodnota 0
echotrain – čas v milisekundách pro potlačení ozvěny, výchozí hodnota 400
readonly – yes/no
trackhook – pouze pro FXO mód, výchozí hodnota no
```

V sekci [general] je možné použít následujících parametrů:

```
format – format dat pro analog (FXS/FXO): alaw/mulaw
idlevalue – hodnota k vyplnění pokud hlasová data chybí; hodnota v intervalu
0-255, standardně 255
buflen – velikost bufferu, standardně 160
errormask – max. možná hodnota 255, ignorováno při typu FXS nebo FXO
priority - výchozí priorita vlákna datové služby; lowest, low, normal,
high, highest
```

2.3 Analogový modul analog.conf

Konfigurační soubor pro správu signalizace a přenosu dat analogových linek odkazuje na výše zmíněný soubor zapcard.conf. Každý řádek zapouzdří

signalizační obvod, který je na rozhraní ovladačů hardwaru používán k odesílání a přijímání událostí (signalizace) a dat (audio). Jsou podporovány dva analogové signalizační typy:

- FXO: Foreign Exchange Office - lokální konec linky je telefonní přístroj
- FXS: Foreign Exchange Station - lokální konec linky je telefonní ústředna

Podobné linky mohou být seskupeny do stejné skupiny se stejným signalizačním typem a parametry [1]. Použití ve volacím plánu je následující:

```
číslo=analog/link/číslo_linky  
^3020$=analog/office-fxo/1
```

Definice analogového linku musí obsahovat následující parametry:

[link] – název linku, na který je odkazováno ve volacím plánu

enable – yes, no

type – FXO nebo FXS

spans – odkaz na spany s analogovými linkami uvedenými v souboru
zapcard.conf

Dále je možno nastavit:

useallcircuits – použít všechny možné okruhy, výchozí hodnota enable

delaydial – zpoždění odesílání čísla na linkách FXO, při jiném typu než FXO je
parametr ignorován, výchozí hodnota 2000

dtmfinband – umožňuje zasílání dat DTMF v pásmu analogových linek, výchozí
hodnota disable

answer-on-polarity – čekání na změnu polarity jako signál zvednutí hovoru na
vzdáleném konci pro linky FXO, výchozí hodnota no

hangup-on-polarity – čekání na změnu polarity při zavěšení hovoru na
vzdáleném konci pro linky FXO, při odchozích hovorech a
zapnutém parametru answer-on-polarity bude první
změna polarity považována za přijetí hovoru, druhá za
zavěšení; při odchozích nebo příchozích hovorech a
vypnutém parametru answer-on-polarity bude

změna polarity považována za událost zavěšení; výchozí hodnota `no`

`callsetup` – parametr se používá k detekci (FXO) nebo zasílání (FXS) dat nastavení linky (např. Caller ID), možné hodnoty:

- `after` – odeslání/detekce mezi prvním a druhým zazvoněním
- `before` – odeslání/detekce před prvním zazvoněním
- `none` – žádná data nejsou odeslána nebo detekována

`debuglevel` – při hodnotě 0 žádné debug zprávy nebudou odesílány

Možnosti nastavení v sekci `[general]`

`echocancel` – automaticky povolit nebo zakázat potlačení echa na začátku/konci hovoru

`call-ended-target` – standardně `tone/busy`

`outoforder-target` – standardně `tone/outoforder`

`call-ended-playtime` – čas pro přehrání parametru `call-ended-target`, po jeho uplynutí se přehraje `outoforder-target`, výchozí hodnota `2000`

`ring-timeout` – lhůta mezi zvoněními příchozích volání na linky FXO, tento časovač může být použit, pokud není žádný způsob detekce uzavření hovoru vzdáleným FXS (beze změny polarity linky), hodnota 0 vypnuto, výchozí hodnota `10000`

`alarm-timeout` – lhůta pro aktivní hovor s linkou v alarmu (out of order), hodnota 0 vypnuto, výchozí `30000`

2.4 Signalizační konfigurační soubor - *ysigchan.conf*

Signalizační trunky je možné definovat pro klasické signalizace na bázi ISDN (QSIG apod).nebo jako SS7 (Signaling System #7).

ISDN trunky se definují pro PRI TDM karty, které mají DAHDI (Zaptel) nebo wanpipe ovladač. Odpovídající moduly v YATE jsou již dříve zmíněné *zapcard.conf* a *wpcard.conf*. Používá se HDLC signalizační rozhraní k odeslání a přijímání dat, které patří do horních vrstev (Q.931, Q.921).

Modul podporuje:

- Manipulaci s SS7 protokolem (řízení, údržbu, směrování)
- Tvorbu/přijem hovorů pomocí ISUP protokolu
- Tvorbu/přijem hovorů pomocí ISDN (Q.931) protokolu
- Pasivní záznam ISDN údajů

Modul pro správné fungování vytváří následující objekty:

- Signalling engine – ke sledování všech signalizačních komponent (circuit groups, call controller, rozhraní apod.)
- SS7 router – identifikuje síť (cestu) k odeslání odchozích SS7 paketů a jejich předávání
- SS7 řídicí objekty pro údržbu a správu SS7 sítě
- Call controller – trunk

Odkaz na definované trunky ve volacím plánu se provádí následovně:

```
volane_cislo=sig/volane_cislo;trunk=nazev_trunku
```

2.4.1 ISDN Trunk

Pro konfiguraci ISDN trunku je zapotřebí definovat následující parametry

[trunk1] – název trunku

type - typ trunku. Možné hodnoty:

- `isdn-bri-net` – ISDN síť pro spojení přes základní HDLC rozhraní
- `isdn-bri-cpe` – ISDN CPE (uživatelská) strana datového spojení přes základní HDLC rozhraní
- `isdn-pri-net` – ISDN síť datového spojení přes jeden nebo více primárních rozhraní HDLC
- `isdn-pri-cpe` – ISDN CPE (uživatelská) strana datového spojení přes jeden nebo více primárních rozhraní HDLC

switchtype – typ přepínače; `euro-isdn-e1`, `euro-isdn-t1`, `national-isdn`, `dms100`, `lucent5e`, `att4ess`, `qsig` nebo `unknown`

sig – odkaz na span obsahující signalizační kanál (D), např. `wanpipe1`

voice – odkaz na span obsahující hlasový kanál (B), např. wanpipe1

offset – standardně 0

Dále může definovaný trunk obsahovat následující parametry:

range - libovolné rozsahy obvodů pro odchozí volání, např. 1,3,5-9

strategy – hodnoty increment, decrement, lowest, highest nebo random

strategy-restrict – hodnoty even, odd, even-fallback nebo odd-fallback

channel-sync – re-synchronizace kanálů v sekundách, min. 60s

channellock – maximální čas (v ms) strávený pokusem uzavřít vzdálený kanál, minimálně 2500ms

numplan – výchozí číslovací plán pro odchozí hovory, hodnoty unknown, isdn, data, telex, national, nebo private

numtype – výchozí typ čísel pro odchozí volání; unknown, international, national, net-specific, subscriber, abbreviated, reserved

presentation – prezentace čísel pro odchozí volání; allowed, restricted, unavailable, výchozí allowed

screening – výchozí „screening“ číslo pro odchozí volání; user-provided, user-provided-passed, user-provided-failed, network-provided, výchozí user-provided

format – formát dat pro odchozí hovory; alaw, mulaw, g721

print-messages – zobraz protokolové datové jednotky na výstupu, standardně no

print-frames – zobraz zprávy „frames“ vrstvy 2 (Q.921) na výstupu

extended-debug – zobraz rozšířené debug data (jako raw hex data) na výstupu, standardně no

2.4.2 Konfigurace SS7

Pro správné funkční nastavení signalizace SS7 je zapotřebí definování ISUP trunku 4. vrstvy, linksetu 3. vrstvy a linku 2. vrstvy SS7.

ISUP trunk musí obsahovat tato nastavení:

[isup1] – název ISUP trunku

type – typ trunku, tedy ss7-isup

pointcodetype – typ SS7 point code, možné parametry

- ITU ITU-T Q.704
- ANSI ANSI T1.111.4
- ANSI8 8-bit SLS
- China GF 001-9001
- Japan T-Q704, NTT-Q704
- Japan5 5-bit SLS

pointcode – místní point code ve formátu network-cluster-member

defaultpointcode – point code pro odchozí hovory

remotepointcode – pointcode pro odchozí zprávy

service – servisní informační pole (číslo protokolu, standardně 5)

priority – priorita v servisním informačním oktetu, hodnota regular,
special, circuit, nebo facility

netindicator – síťový indikátor v SIO, hodnoty international,
spareinternational, national, reservednational

lockgroup – povolí zasílání požadavků pro block/unblock

earlyacm – převod příchozích ACM uživatelských stavů do progres nebo vyzvánění

voice – specifikace spanů pro hlasové kanály, možno oddělovat čárkou, např.
wanpipe1, wanpipe2

offset – standardně 0

strategy - použitá strategie k přidělení hlasových kanálů pro odchozí hovory,
možné hodnoty, standardně increment:

- increment – vzestupné vyhledávání nečinného kanálu začínající s posledním alokovaným kanálem
- decrement – sestupné vyhledávání nečinného kanálu začínající s posledním alokovaným kanálem
- lowest – vzestupné vyhledávání nečinného kanálu začínající prvním kanálem ve skupině
- highest – sestupné vyhledávání nečinného kanálu začínající posledním kanálem ve skupině
- random – náhodné zvolení nečinného kanálu

strategy-restrict – omezení a chování alokace:

- even – omezení na sudé kanály
- odd – omezení na liché kanály
- even-fallback – alokace sudých kanálů, fall back na liché
- odd-fallback – alokace lichých, fall back na sudé

chansync – re-synchronizace kanálů v sekundách, min. 60s

netind2pctype – např. ITU

presentation – prezentace čísel pro odchozí volání; allowed, restricted, unavailable, výchozí allowed

screening – „screening“ pro odchozí volání; user-provided, user-provided-passed, user-provided-failed, network-provided, výchozí user-provided

inn – směrování na interní číslo sítě, výchozí hodnota no

format – formát dat pro odchozí hovory; alaw, mulaw, g721

Dále je možno nastavit:

numplan – výchozí číslovací plán pro odchozí hovory, hodnoty unknown, isdn, data, telex, national, nebo private

numtype – výchozí typ čísel pro odchozí volání; unknown, international, national, net-specific, subscriber, abbreviated, reserved

print-messages – zobraz protokolové datové jednotky na výstupu, standardně no

extended-debug – zobraz rozšířené debug data (jako raw hex data) na výstupu

userparttest – hodnota v sekundách

parttestmsg – ISUP zpráva pro part test, standardně UPT

range - libovolné rozsahy obvodů pro odchozí volání, např. 1,3,5-9

Nastavení linksetu 3. vrstvy

[linkset1] – název linksetu

type – hodnota ss7-mtp3

netind2pctype – point code typy, používané k mapování příchozích síťových indikátorů, jedna hodnota nebo 4 hodnoty oddělené čárkou, např.

ITU **nebo** ANSI8, ITU, ANSI, ITU

`adjacent` – vytvoří přilehlou trasu pro SS7, hodnota ve formátu:

`pointcodetype, label`

`local` – lokální `pointcode`; `pointcodetype, label`

`link` – název linku pro SS7 vrstvu 2, parametr může být opakován pro vytvoření více linků

`autostart` – standardně `yes`

`link1.sig` – odkaz na span se signalizačním kanálem

`emergency` – `yes/no`

Dále je možno nastavit:

`netindicator` – síťový `indicator`; `international, spareinternational, national, nebo reservednational`

`route` – vytvoří cíl trasy pro SS7 síť, hodnota ve formátu:

`pointcodetype, label, priority, např. ITU, 2-2-2, 100`

`allowed` – explicitní povolení `pointcode`, při neuvedení povoleny všechny dostupné trasy

`checklinks` – standardně `true`

`forcealign` – přeskupí linky, které již nereagují na SLTM; `true` nebo `false`

`checkfails` – interval v ms pro znovu odeslání SLTM zpráv (Q.707 T1)

`maintenance` – interval v ms pro odeslání SLTM zpráv (Q.707 T2)

Nastavení linky 2. vrstvy:

`[link1]` – název linku

`type` – v tomto případě `ss7-mtp2`

`autostart` – standardně `yes`

`emergency` – nouzové přizpůsobení SS7 MPT2 vrstvy při startu

`filllink` – opakování paketu při odeslání FISU nebo LSSU paketů

`rxunderrun` – maximální interval v ms mezi dvěma pakety při detekci běhu

`underrun`, výchozí hodnota 0 (disable)

3. Srovnání PBX Yate s PBX Asterisk

Přestože se může zdát, že pobočkové ústředny Yate a Asterisk jsou si podobné, sami vývojáři na svých stránkách toto tvrzení popírají.

Asi nejvýraznější rozdíl mezi PBX Yate a PBX Asterisk je v registraci uživatelů. Zatímco v PBX Asterisk je zapotřebí definovat účet v konfiguračním souboru pro určitý protokol a ve volacím plánu, Yate použitý protokol nerozlišuje a pro registraci používá jediný soubor, jak znázorňuje následující tabulka:

PBX Asterisk	PBX Yate
sip.conf	regfile.conf
[101] type=friend secret=1234 Host=dynamic	[101] password=1234
Iax.conf	[102] password=1234
[102] type=friend secret=1234 Host=dynamic	
extensions.conf	
[default] exten => 101,1,Dial(SIP/101) exten => 102,1,Dial(IAX2/102)	

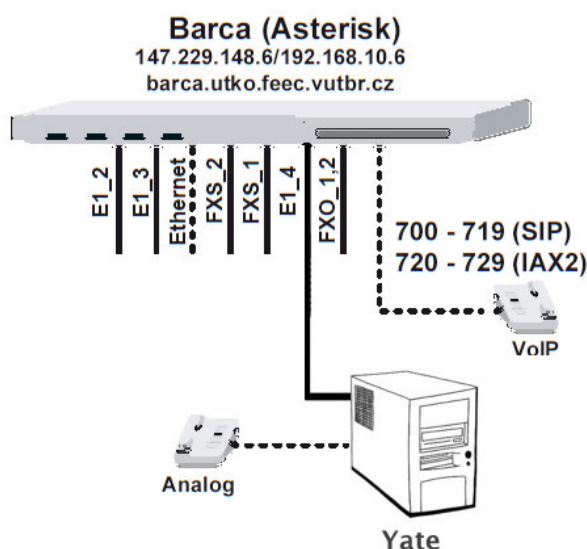
Tabulka 2: Registrace uživatelů 101 pro SIP a 102 pro IAX
z pohledu jednotlivých ústředen

Rozdílná je i syntaxe zápisu směrovacích pravidel. Zatímco pro Asterisk je pro volání zapotřebí použití klíčového slova *exten* a poté zápis destinace pro danou klapku, v Yate se směrovací pravidlo zpracovává formou „rovnice“, tedy klíčovým znakem je znak „ = “ (rovná se). Zadaná volba je porovnána s levou stranou rovnice a pokud zápis odpovídá zadané volbě, provede se příkaz na pravé straně rovnice.

Pobočková ústředna Yate je v mnoha ohledech podobná pobočkové ústředně Asterisk. Stejně jako Asterisk má Yate několik dílčích konfiguračních souborů, v mnoha případech s podobnými parametry nastavení. Oproti Asterisku má však na první pohled větší možnosti nastavení (např. použití H.323 gateway i gatekeperu apod.), na druhou stranu je popis konfiguračních souborů ze strany vývojářů velmi slabý. Použité parametry a možnosti použití jsou popisovány především teoreticky, často chybí praktické ukázky použití pro různé aplikace.

4. Praktická část

V praktické části byla v počítačové učebně nainstalována a nakonfigurována ústředna YATE. Byli definováni uživatelé 101 a 102 (*regfile.conf*), nakonfigurovány a odzkoušely protokoly SIP (*ysipchan.conf*), IAX (*yiaxchan.conf*) a H.323 (*h323chan.conf*) a to jak pomocí softwarových telefonů Vopier, 2N Lite, YateClient, tak i pomocí hardwarových telefonů Well 3130F a Interbell voláním na klapky 101 a 102, analogové klapky 103 a 104 a SIP účty na serveru Barca 710 a 760.



Obr. 4 – Schéma zapojení učebny

Počítač je vybaven hardwarovou kartou E1/T1 Digium TE120 a analogovou kartou Sangoma A200. Pro kartu Digium byly nastaveny trunky (*zapcard.conf*) pro připojení E1/T1 k ústředně Asterisk (Barca):

- ISDN PRI QSIG - hlasové kanály 1-15 a 17-31, signalizační kanál 16
- SS7 – hlasové kanály 2-31, signalizační kanál 1

Dále byly definovány trunky na analogové linky pomocí ovladačů DAHDI (emulované kanály 34 a 35, *zapcard.conf*), neboť Yate nepodporuje přímé použití analogových karet Sangoma. Na tyto analogové linky jsme ve volacím plánu přiřadili telefonní čísla 103 a 104.

Kromě volání mezi telefony jsme také odzkoušeli funkce pro konferenční volání, call fork, tónový generátor a přehrávání a záznam audio souborů.

Nastavení jednotlivých konfiguračních souborů a výpis debug zpráv pro jednotlivá volání a signalizace jsou uvedeny v příloze 1 a 2.

5. Zadání úlohy 1 – Seznámení s PBX YATE a jeho základní konfigurace

► **Úkol 1.1 – Seznamte se strukturou pracoviště s PBX YATE. Prostudujte konfigurační soubory, zejména pak `ysipchan.conf`, `regfile.conf`, `yate.conf`, `system.conf` a `yiaxchan.conf`.**

Všechny uvedené soubory naleznete na následujících odkazech:

- `/etc/yate/ysipchan.conf`
- `/etc/yate/regfile.conf`
- `/etc/yate/yate.conf`
- `/etc/dahti/system.conf`
- `/etc/yate/yiaxchan.conf`

► **Úkol 1.2 – Nastavte výpis debug zpráv pro použité protokoly SIP, IAX a H.323, spusťte vaši pobočkovou ústřednu YATE s barevně odlišeným výpisem zpráv, prozkoumejte jednotlivé výpisy zpráv z načítaných modulů. Vyzkoušejte klávesové zkratky pro reload a ukončení YATE.**

Výpis debug epote pro jednotlivé moduly nastavíte v konfiguračním souboru `yate.conf` v adresáři `/etc/yate` v sekci `[debug]` přidáním řádku ve tvaru

```
název_modulu=level číslo
```

```
[debug]
```

tedy např:

```
iax=level 10
```

```
sip=level 10
```

Spuštění YATE s výpisem zpráv provedete z příkazové řádky terminálu zadáním příkazu

```
yate -vvvv -Do
```

příčemž počet použitých písmen „v“ ovlivňuje míru zobrazovaných informací. Reload provedete pomocí klávesové zkratky „CTRL + \“, ukončení YATE pak kombinací kláves „CTRL + C“

► **Úkol 1.3 – Prostudujte nastavení konfiguračního souboru pro epote manager – CLI (Command Line Interface) – pro YATE, připojte se přes telnet k CLI, prozkoumejte možnosti příkazů CLI a vyzkoušejte restart a reload YATE přes CLI.**

IP adresa pro epote manager je nastaven na lokální smyčku, tedy 127.0.0.1, na portu 5038, připojení přes telnet pak provedete např. otevřením dalšího terminálového okna a z jeho příkazové řádky následovně:

```
telnet localhost 5038
```

Výpis možných příkazu provedete zadáním volby `help` a restart resp. reload YATE pak stejnojmenným příkazem. Připojení epote manageru můžete také ověřit v okně terminálu s YATE, který vypíše

```
epote connection from 127.0.0.1:56037 to 127.0.0.1:5038
```

Nezapomeňte po každé změně konfigurace YATE restartovat.

► **Úkol 1.4 – Pomocí parametrů v konfiguračním souboru pro SIP nakonfigurujte 2 softwarové telefony tak, aby používaly SIP protokol, registrovaly se do vaší pobočkové ústředny pod účtem 101 a 102, hesla zvolte shodné s číslem účtu pro danou klapku. Ověřte registraci testovacím hovorem.**

Na pracovišti je nainstalovaný softwarový telefon Zopier, který umožňuje registraci více účtu najednou, nicméně můžete použít i jiné softwarové telefony, např. YATE Client, který má jednoduché administrační rozhraní. Pro program Zopier postupujte následovně:

1. Pomocí ikony na ploše spusťte program
2. Použijte klávesovou zkratku „CTRL + O“ nebo klikněte pravým tlačítkem myši a zvolte možnost „Options“

3. Vytvořte SIP účet pomocí „SIP Accounts“ → „Add new SIP account“. IP adresu zvolte dle IP uvedené v konfiguračním souboru, *Username* a *Password* zvolte dle čísla klapky, kterou definujete. Caller ID Name je identifikace volajícího. Jednotlivé klapky 101 a 102 najdete nadefinované s konfiguračním souboru `regfile.conf` adresáři `/etc/yate`.
4. Samotná registrace a odregistrace se provádí pomocí tlačítka Register, resp. Unregister.

Obdobným způsobem zaregistrujte i druhou klapku na druhém VoIP softwarovém telefonu, ověřte registraci voláním mezi softwarovými telefony voláním na klapku 102 z prvního telefonu a na klapku 101 z druhého telefonu.

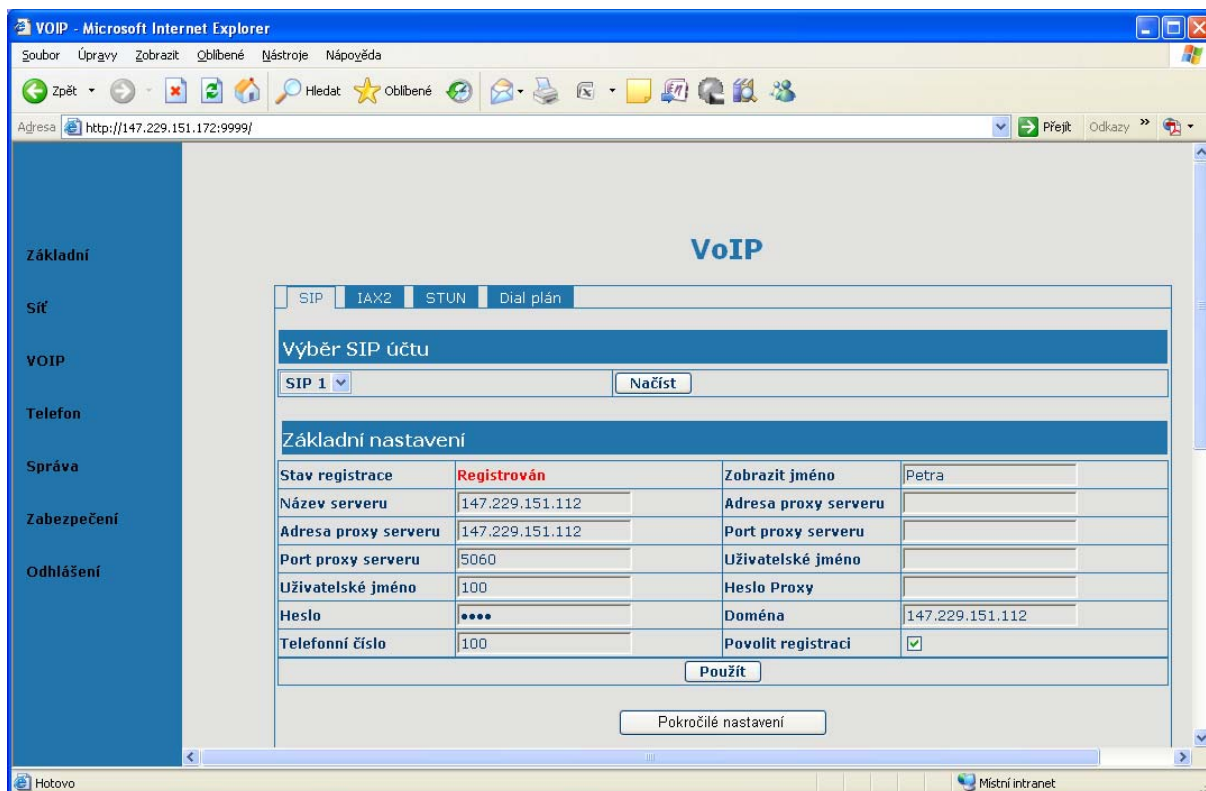
► Úkol 1.5 – Obdobným způsobem jako v úkolu 1.4 nakonfigurujte softwarové telefony tak, aby používaly IAX protokol, registrovaly se do vaší pobočkové ústředny pod účtem 101 a 102, hesla zvolte shodné s číslem účtu pro danou klapku. Ověřte registraci testovacím hovorem.

YATE v registračním souboru `regfile.conf` nerozlišuje použité protokoly pro definované klapky. Odregistrujte tedy obě SIP klapky na softwarových telefonech, definujte nové IAX účty 101 a 102 (demoverze Zoiperu umožňuje pouze 3 definované účty, můžete tedy např. SIP 102 účet vymazat pravým tlačítkem myši) a zaregistrujte obě klapky softwarovým telefonem. Ověřte testovacími hovory na obě klapky. Obdobně také můžete volat vzájemně mezi protokoly, např. zaregistrováním účtu 101 na SIP a 102 na IAX.

► Úkol 1.6 – Připojte se na ethernetové rozhraní VoIP telefonu Well 3130F a nastavte ho tak, aby používal protokol SIP a IAX. Použijte stejné přihlašovací údaje stejné z předchozího úkolu 1.4, 1.5. Ověřte registraci testovacím hovorem.

- odregistrujte všechny vámi zaregistrované účty na softwarových telefonech
- v terminálové konzoly OS Linux zjistěte IP adresu vašeho počítače pomocí příkazu `ifconfig`

- zmáčkněte tlačítko *Local IP/OK* na VoIP telefonu Well 3130F a vyčtěte z displeje přiřazenou IP adresu telefonu
- spusťte www prohlížeč (např. Mozilla Firefox) a do prohlížeče zadejte IP adresu vašeho telefonu, port 9999. (např. 192.168.1.161:9999)
- v prohlížeči se objeví formulář pro zadání přihlašovacích údajů do VoIP telefonu Well 3130F. Zadejte login a heslo (pokud nevíte, obraťte se na vyučujícího nebo správce učebny). Pro přihlášení si prostudujte jednotlivé položky v menu.
- zvolte položky menu VOIP a podmenu SIP nebo IAX2. Pokud chcete nastavit telefon pro SIP účet, je nutné, aby v podmenu IAX2 položky *Povolit registraci* a *IAX2 (výchozí protokol)* byly odškrtnuty, podobně pokud chcete nastavit telefon pro IAX účet, odškrtněte v podmenu SIP položku *Povolit registraci*
- nastavte v podmenu SIP nebo IAX všechny potřebné údaje pro registraci telefonu k ústředně PBX YATE stejně jako v předchozích úlohách pro klapku 101
- proces registrace VoIP telefonu do PBX Asterisk provedete zaškrtnutím *Povolit registraci* pro SIP a pro IAX navíc ještě zaškrtnutím *Použít jako výchozí protokol*. Stav registrace by měl být *Registrován*. Při špatném nastavení Uživatelského jména nebo hesla se objeví chybová zpráva *Failed with 403*
- nepoužívejte na VoIP telefonu více jak jeden účet zároveň
- v softwarovém telefonu nadefinujte a registrujte klapku 102 (viz úkol 1.3, 1.4) a prověřte vzájemné volání mezi softwarovým a VoIP telefonem Well 3130F pro různé kombinace použitých protokolů SIP a IAX



Obr. 5 – Prostředí konfiguračního rozhraní telefonu Well 3130F

► **Úkol 1.7 – Seznamte se s konfiguračním souborem pro registraci uživatelů regfile.conf a upravte jej tak, aby identifikace volajícího pro klapku 101 byla nastavena na jméno *Jarda*.**

Identifikace volajícího je nastavena parametrem *username*. Změňte tedy tento řádek na *username=Jarda*. Ověřte zaregistrováním této klapky, tedy 101, jako SIP účet na softwarovém telefonu, na VoIP telefonu Well zaregistruje SIP účet 102 a ověřte voláním na 102 na displeji telefonu. Nezapomeňte restartovat YATE.

► **Úkol 1.8 – Vytvořte další 2 účty pro registraci 1100 a 1200. Pro účet 1100 zvolte heslo 1100, kodek alaw a jméno Mirek, pro druhý účet 1200 přiřadte jméno Standa, formát plaw a heslo zvolte 1200. Pro oba účty použijte kodek alaw. Ověřte účty zaregistrováním telefonů.**

Obdobně jako již přednastavené účty, vytvoříte účty tak, že název účtu (volací číslo) umístíte do hranatých závorek, heslo nastavíte parametrem *password* a jméno účtu (zobrazovaný název) parametrem *username*. Nastavené účty pak budou zapsány např. následujícím způsobem:

```
[1100]
password=1100
username=Mirek
formats=alaw
```

```
[1200]
password=1200
username=Standa
formats=ulaw
```

► **Úkol 1.9 – Seznamte s konfiguračním souborem pro H.323, nastavte použití protokolu H.323 a YATE jako gateway, použijte softwarový telefon 2N Star, který podporuje protokol H.323, a ověřte voláním.**

Konfigurační soubor YATE *h323chan.conf* pro H.323 najdete v adresáři */etc/yate*. Obsahuje základní sekce pro nastavení H.323 a to *[generals]* pro obecné nastavení, *[codecs]* pro nastavení kodeků, *[ep]* pro nastavení jako endpoint a *[gk]* pro použití jako gatekeeper. V sekci pro použité kodeky zakažte parametr *default* a povolte kodeky *alaw* a *gsm*, tedy:

```
[codecs]
default=disable
alaw=enable
gsm=enable
```

V sekci *[ep]* povolte použití jako endpoint (parametr *ep*) a gateway (parametr *gw*), nastavte *bind* adresu na lokální IP adresu počítače, *port* zvolte standardně 1720 a povolte H.245 tunelování (parametr *h245tunneling*) a *faststart*, tedy:

```
[ep]
ep=true
gw=true
addr=192.168.10.174
port=1720
```

```
faststart=true  
h245tunneling=true
```

V softwarovém telefonu 2N Star vytvořte H.323 účet, zvolte způsob spojení *H.323 gateway*, nastavte adresu serveru a port shodný s vaším konfiguračním souborem, stejně tak zvolte parametry *Faststart* a *H.245 tunneling*. Ověřte voláním na jednu z registrovaných klapků.

► **Úkol 1.10 – Prozkoumejte v terminálu se spuštěným YATE H.323 zprávy, upravte úroveň debug levelu pro H.323 a zapněte externí YATE RTP modul. Ověřte funkčnost voláním**

Přestože jste již nastavovali debug level v hlavním konfiguračním souboru *yate.conf*, můžete dodatečně nastavit debug level v sekci *general* souboru *h323chan.conf* parametry *debug* a *dumpcodecs* pro použité kodeky. Externí RTP modul zapnete parametrem *external_rtp*, tedy:

```
[general]  
debug=3  
dumpcodes=1  
external_rtp=yes
```

Nastavení RTP modulu prozkoumejte v konfiguračním souboru *yrtchan.conf*.

6. Zadání úlohy 2 – Volací plán a PBX pomocné funkce

- **Úkol 2.1 – Seznamte se s číslovacím plánem, způsobem zápisu, práci s kontexty, parametry a proměnnými viz kapitola XXX**

Soubor s konfigurací volacího plánu najdete v `/etc/yate` v souboru `regexroute.conf`

- **Úkol 2.2 – Definujte v defaultním kontextu směrovací pravidla tak, aby při volání jakéhokoliv čísla začínajícího na 1 byl volán registrovaný uživatel 101 a při volání jakéhokoliv čísla začínajícího na 2 (2XXX) byl volán registrovaný uživatel s číslem následujícím po čísle 2 (tedy XXX), tzn. aby při volání např. 2345 byl volán registrovaný uživatel 345.**

Rozdělení volaného výrazu můžeme rozdělit do podvýrazů pomocí znaků `\(` a `\)` a můžeme pak na něj odkazovat pomocí `\číslo_podvýrazu` viz kapitola X.X. Nahrazení parametru volaného čísla (destinace) můžeme provést pomocí parametru `called=nahrazující_výraz`. Zápis zadaného příkladu může vypadat následovně:

```
[default]
^1\(.*\)$=return;called=101
^2\(.*\)$=return;called=\1
```

Znak `^` značí začátek řetězce, znak `$` pak konec řetězce.

- **Úkol 2.3 – Obdobným způsobem nadefinujte pravidlo pro volání začínající na číslo 3. Vytvořte vlastní kontext s názvem `kontext1`, na který odkážete z defaultního kontextu. V tomto vámi definovaném kontextu vytvořte pravidlo, podle kterého bude ze zadaného řetězce znaků voláno na číslo bez posledního zadaného čísla, tedy dle vzoru `3XXXXY → 3XXXX`**

Odkaz na kontext ze zadaného pravidla můžeme psát přímo za znak `=`, zápis tedy může vypadat následovně:

```
[default]
^3\(.*\)$=kontext1
```

```
[kontext1]
^3\(...*\)\(...*)$=return;called=3\1
```

Znaky \1 značí první podvýraz. Např. pro volbu 3114 uvedené pravidlo vrátí 311. Číslo 11 je uloženo v podvýrazu \1, číslo 4 v podvýrazu \2 a celé číslo 3114 bychom mohli odkazovat pomocí volání celého výrazu tedy \0. Tohoto pravidla můžeme s výhodou použít při volání na YATE z jiné ústředny při použití signalizace SS7, které standardně posílají ukončovací znak #. Nezapomeňte po každé změně konfigurace YATE restartovat.

► **Úkol 2.4 – Vytvořte pravidlo pro volání na konferenční hovory pro číslo 4444 a 4445 tak, aby každé číslo mělo svoji vlastní konferenční místnost. Ověřte testovacím hovorem ze 2 různých telefonů.**

Vytvoření tzv. konferenční místnosti provedete opět ve volacím plánu viz kapitola X.X. Definování 2 konferenčních místností pak může vypadat následovně:

```
[default]
^4444$=conf/mistnost1
^4445$=conf/mistnost2
```

► **Úkol 2.5 – Vyzkoušejte si záznam a přehrávání audiosouborů. Nahrávání definujte pro číslo 9999, přehrávání záznamu pro číslo 9998.**

Jak je popsáno v kapitole X.X, záznam a přehrávání můžeme definovat ve směrovacím plánu následovně:

```
[default]
^9999$=wave/record//home/user/zaznam.wav
^9998$=wave/play//home/user/zaznam.wav
```

Cestu pro soubor volte tak, abyste měli právo zápisu. Mezi parametrem rozlišujícím záznam nebo přehrávání a cestou k souboru musí být vždy dvě lomítka, tedy // .

► **Úkol 2.6 – Vytvořte pravidlo pro volání více alternativním registrovaným uživatelům zároveň. Při volbě čísla 5555 dojde k volání uživatele 101 a 102 zároveň. Při volbě 5556 dojde nejprve k volání uživatele 101, po 5 sekundách dojde i k volání uživatele 102 (volání na 101 stále pokračuje) a po dalších 5 sekundách se volání na obě čísla ukončí.**

Volání na více čísel zároveň řeší funkce call fork viz kapitola X.X. Pro volání registrovaných uživatelů se používá parametru lateroute, zápis pak může vypadat následovně:

```
[default]
^5555$=fork lateroute=101 lateroute=102
^5556$=fork lateroute=101 |next=5000 lateroute=102 |exec=5000
```

Parametr *next=5000* způsobí vytáčení po uplynutí zadaného času v milisekundách, pokud bychom použili parametr *drop*, došlo by k ukončení volání předchozího volaného čísla. Doporučuje se řádek s funkcí call fork ukončovat řádkem

```
;stoperror=busy.
```

► **Úkol 2.7 – Seznamte se s konfiguračními soubory pro signalizaci a konfiguraci analogových a digitálních E1/T1 karet Digium a Sangoma.**

Soubory pro signalizaci a nastavení digitálních E1/T1 a analogových karet jsou následující:

- */etc/yate/wpcard.conf* pro nastavení hlasových a signalizačních kanálů pro E1/T1 karty Sangoma
- */etc/yate/zapcard.conf* pro nastavení hlasových a signalizačních kanálů pro E1/T1 a analogové karty Digium
- */etc/yate/analog.conf* pro vytvoření analogových spanů na které se odkazuje ve volacím plánu
- */etc/yate/ysigchan.conf* pro vytvoření E1/T1 spanů na které se odkazuje ve volacím plánu

V souboru *zapcard.conf* si prohlédněte E1 span definovaný pro hlasové kanály 1-15 a 17-31 a 16 kanál signalizace pro Q-sig, resp. 2-31 kanál pro hlas a kanál 1 pro signalizaci SS7 pojmenovaný *span1*, v souboru *ysigchan.conf* pak definovaného

linku s názvem *link1*, ve kterém se odkazuje na hlasové a signalizační kanály přes výše zmíněny *span1*, a ve směrovacím plánu *regexroute.conf* použití směrovacího pravidla na tento link pomocí syntaxe

```
zadane_cislo=sig/volane_cislo;trunk=link1
```

► **Úkol 2.8 – Vytvořte span pro definici hlasových analogových okruhů typu FXS a analogový link odkazující na vámi definovaný span.**

YATE umožňuje vytváření analogových okruhů pouze pro karty Digium. Přestože je ve vašem PC nainstalována analogová karta Sangoma, jsou její ovladače emulovány přes DAHDI a můžete odkazovat na analogové okruhy stejně jako na digitální E1/T1. Po prostudování souboru */etc/dahdi/system.conf* zjistíte, že analogové okruhy jsou definovány na kanálech 34 a 35 (řádky *fxoks*). V souboru *zapcard.conf* tedy vytvořte např. spany *[span2]* a *[span3]* pro každý analogový kanál zvlášť typu FXS:

```
[span2]
type=FXS
offset=0
voicechans=34
```

```
[span3]
type=FXS
offset=0
voicechans=35
```

Analogový link vytvoříte v souboru *analog.conf*. Vytvořte např. názvy linků *[ana-link1]* a *[ana-link2]*, nastavte typ FXS a odkažte je na vámi vytvořené spany:

```
[ana-link1]
enable=yes
type=FXS
spans=span2
```

```
[ana-link2]
enable=yes
type=FXS
spans=span3
```

► **Úkol 2.9 – Vytvořte směrovací pravidlo tak, aby pro čísla začínající na 103 a 104 došlo k volání na analogové linky. Ověřte voláním na analogový telefon.**

V předchozích úlohách jste vytvořili analogový link *link1* odkazující na *span2* a *link2* na *span3*. Nyní již stačí vytvořit směrovací pravidlo ve formátu *cislo=analog/link* tedy:

```
^103\(.*\)$=analog/link1
^104\(.*\)$=analog/link1
```

► **Úkol 2.10 – Seznamte se s další dokumentací pro pobočkovou ústřednu YATE uvedenou v sekci Reference viz [1].**

Závěr

V práci bylo pojednáno o pobočkové ústředně Yate (kap. 1), způsob její instalace na linuxové distribuci Ubuntu (kap. 1.1), byly popsány její vlastnosti, architektura a kodeky (kap. 1.2), které podporuje. Dále pak byly popsány jednotlivé dílčí konfigurační soubory pro základní fungování jednotlivých modulů ústředny SIP (kap. 1.3), IAX (kap. 1.4) a H.323 (kap. 1.5). Byl popsán způsob registrace uživatelů do ústředny a způsob práce s nimi (kap. 1.7), způsob směrování ve volacím plánu pomocí regulárních výrazů (kap. 1.6) včetně příkladů použití (kap. 1.6.2). Práce dále popisuje, jaké hardwarové karty ústředna Yate podporuje (kap. 2), jakým způsobem je zapotřebí nakonfigurovat jednotlivé trunky (kap. 2.1 a 2.2) pro připojení analogových telefonů (kap. 2.3) a digitální ústředny (kap. 2.4) pomocí signalizace QSIG (kap. 2.4.1) a SS7 (kap. 2.4.2). Získané poznatky pak byly aplikovány v počítačové učebně na PC s Yate připojeným přes E1 port k ústředně Asterisk (Barča) a analogovým portem k telefonu Interbell (kap. 4). Funkčnost byla ověřena voláními mezi ústřednami a telefony (Příloha 1 a 2)

Dále byla ústředna Yate porovnána s ústřednou Asterisk a byly zdůrazněny její výhody i nevýhody (kap. 3). Výsledkem práce jsou pak dvě vzorové laboratorní úlohy (kap. 5 a 6) zabývající se základním nastavením Yate, jednotlivých protokolů SIP, IAX a H.323, volacím plánem a definováním jednotlivých trunků pro volání mimo ústřednu a opačně.

Yate lze považovat za plnohodnotného nástupce velmi rozšířené a populární pobočkové ústředny Asterisk. Podobně jako Asterisk umožňuje libovolné nastavení funkčních aplikací. Nevýhodou Yate je však již dříve zmíněná špatná úroveň dokumentace, která zapříčiňuje, že pro obsáhlejší a složitější řešení je uživatel nucen zdlouhavě hledat řešení z různých informačních zdrojů zveřejněných jinými uživateli, kterých je, vzhledem k tomu, že je Yate málo rozšířen oproti např. Asterisku, velmi málo.

Seznam literatury

- [1] TEAM, Null. *YATE* [online]. 2008, 2011 [cit. 2011-05-25]. Dostupné z WWW: <<http://yate.ro/pmwiki/index.php>>.
- [2] JOHNSTON, Alan B. . *SIP: understanding the Session Initiation Protocol*. [s.l.] : [s.n.], 2009. 395 s.
- [3] PUŽMANOVÁ, Rita. *Lupa.cz* [online]. 11.11.2004 [cit. 2011-05-25]. Protokol SIP ve zkratce. Dostupné z WWW: <<http://www.lupa.cz/clanky/protokol-sip-ve-zkratce/>>.
- [4] BOUCADAIR, Mohamed. *Inter-asterisk exchange (IAX) deployment scenarios in SIP-enabled networks*. [s.l.] : [s.n.], 2009. 251 s.
- [5] SEGREST, Rick. *Networkworld* [online]. 2005 [cit. 2011-05-25]. IAX protocol simplifies VoIP . Dostupné z WWW: <<http://www.networkworld.com/news/tech/2005/041105techupdate.html>>.
- [6] *International Telecommunication Union* [online]. 2009 [cit. 2011-05-25]. H.323 : Packet-based multimedia communications systems. Dostupné z WWW: <<http://www.itu.int/rec/T-REC-H.323/e>>.
- [7] BINDER, Bc. Tomáš. *SPRÁVA A KONFIGURACE VOIP ÚSTŘEDNY ASTERISK*. Brno, 2008. 57 s. Diplomová práce. VUT v Brně.
- [8] LUDÍK, Martin. *Domácí VoIP ústředna s připojením do GSM sítí*. Zlín, 2009. 57 s. Bakalářská práce. UTB ve Zlíně.
- [9] *Digium* [online]. 2006 [cit. 2011-05-25]. Dostupné z WWW: <<http://www.digium.com/en/>>.
- [10] *Sangoma* [online]. 2005 [cit. 2011-05-25]. Dostupné z WWW: <<http://sangoma.com>>.
- [11] BÍBR, Ivan. *Ubuntu 10.04 CZ: praktická příručka uživatele Linuxu*. [s.l.] : Computer Press, 2010. 366 s.

Seznam použitých zkratek

DTMF	Dual-tone multi-frequency
HDLC	High-Level Data Link Kontrol
IAX	Inter-Asterisk eXchange
ISDN	Services Digital Network
IVR	Interactive Voice Response
SIP	Session Initiation Protocol
SS7	Signaling system #7
PBX	Private branch exchange
PTSN	Public switched telephone network
VoIP	Voice over Internet Protocol

Seznam obrázků

Obr. 1 – Diagram architektury YATE

Obr. 2 – Blokové schéma provázanosti jednotlivých modulů

Obr. 3 – Znázornění modulů YATE jako funkčních bloků vzhledem k HW kartám

Obr. 4 – Schéma zapojení učebny

Obr. 5 – Prostředí konfiguračního rozhraní telefonu Well 3130F

Seznam tabulek

Tabulka 1: Seznam podporovaných kodeků

Tabulka 2: Registrace uživatelů 101 pro SIP a 102 pro IAX z pohledu jednotlivých ústředen

Seznam příloh

Příloha 1 – Konfigurační soubory použité v praktické části

Příloha 2 – Výpis debug zpráv pro testovací volání

Příloha 3 – výpis použitelných funkcí v souboru regexroute.conf

Příloha 4 – Možnosti parametrů pro spuštění Yate

Příloha 1 – Konfigurační soubory použité v praktické části

accfile.conf

```
[test_sip]
enabled=yes
protocol=sip
description=Test SIP account
formats=alaw,mulaw,gsm

[test_h323]
enabled=yes
protocol=h323
description=Test H.323 account

[test_iax]
enabled=yes
protocol=iax
description=Test IAX2 account
```

analog.conf

```
[ana-link1]                ;navez linku
enable=yes
type=FXS
spans=span2                ;odkaz na spany v souboru zapcard.conf
[ana-link2]                ;navez linku
enable=yes
type=FXS
spans=span3                ;odkaz na spany v souboru zapcard.conf
```

h323chan.conf

```
[general]
debug=3                    ;debuglevel
dumpcodecs=0               ;deublevel pro kodeky, vypnuto
dtmfband=no                ;detekce DTMF na vstupu
external_rtp=yes           ;pouziti modulu yrtchan.conf

[codecs]
```

```

default=disable
alaw=enable
gsm=enable
[ep]
ep = true           ;pouziti jako endpoint
gw = true           ;deklarovat se jako gateway
addr=192.168.10.174 ;bind adresa
port=1720
faststart=true      ;povoleni metody faststart
h245tunneling=true  ;povoleni metody H.245 tunneling
h245insetup=false   ;nepovoluje se pro zapnuty faststart
dtmfband = false    ;detekce DTMF na vstupu
silencedetect = none ;detekce ticha
[incoming]           ;sekce pro prichozi volani
context=default      ;vychozi kontext
maxqueue=5           ;max.delka fronty
called=101           ;volane cislo pokud není uvedene ve volacím planu

```

pbxassist.conf

```

[general]
enabled=yes
priority=15
default=yes
incoming=yes         ;pouzit PBX assist pro prichozi hovory
timeout=2000
lang=cz

```

regexroute.conf

```

[contexts]
${id}^sig/==ss7_kon ;pro signalizaci SS7 skok do kontextu ss7_kon
${address}^ana-link1/34=return;caller=103 ;prirazeni cisla 103 pro analog. linku 1
${address}^ana-link2/35=return;caller=104 ;prirazeni cisla 104 pro analog. linku 2
.*=default          ;skok do kontextu default

```

[default]

```
.*=echo caller: ${caller} called: ${called} id: ${id} address: ${address} status ${status}
; pro všechny volby vypis parametru, pro testovani
^off-hook$=external/nodata/overlapped.php;accept_call=true
;Pro analogove telefony overlapped.php - v pbxassist.conf musi byt dtmfpass=yes
;vse co je do polozky ${overlapped} se uplatnuje pri overlapp volbe - dulezita je
;pevna delka
^103$=analog/ana-link1 ;pro volbu 103 uskutecni volani přes analogovu linku 1
^104$=analog/ana-link2 ;pro volbu 104 uskutecni volani přes analogovu linku 2
^[2-8]\(..)$=sig^0;trunk=link2 ;pro volbu [2-8]XX uskutecni volani ven z ustredny
${overlapped}yes=return;error=incomplete
```

[ss7_kon]

```
^103\(..)$=analog/ana-link1 ;pro cisla zacinajici 103 → analogova linka 1
^104\(..)$=analog/ana-link2 ;pro cisla zacinajici 104 → analogova linka 1
^1\(..)\(..)$=return;called=1\1 ;pro prichizi volani 1XX z vnejsku, nebot pro SS7 je
;volba zakoncena znakem #, který vsak Yate nedokáže odfiltrvat a pak dochází ke
;špatné interpretaci volby, tedy pro volbu 101 Asterisk odesle 101#, Yate pak zachytí
; do subvyrazu \1 volbu 01 a krizek do subvyrazu \2
```

regfile.conf

[general]

```
autocreate=yes ;vytvorit vstupy pro uzivatele
auth=100 ;nastaveni priorit
register=100
route=100
preroute=100
```

[101]

```
enabled=yes
username=101
password=101
```

[102]

```
enabled=yes
username=102
password=102
```

tonegen.conf

[general]

lang=cz

[cz]

dial=425/330,0/330,425/660,0/660

busy=425/330,0/330

ring=425/1000,0/4000

congestion=425/165,0/165

callwaiting=425/330,0/9000

dialrecall=!425/100,!0/100,!425/100,!0/100,!425/100,!0/100,425/330,0/330,425/660,0/660

record=1400/500,0/14000

info=950/330,0/30,1400/330,0/30,1800/330,0/1000

stutter=425/450,0/50

yate.conf

[general]

modload=enable ;nacteni vsech modulu

yiaxchan.conf

[general]

port=4569

addr=192.168.10.174 ;bind adresa

[formats]

default=disable

preferred=alaw ;preferovany format

mulaw=enable

alaw=enable

yrtchan.conf

[general]

minport=16384 ;rozsah portu

maxport=32768

localip=192.168.10.174
tos=mincost ;"type of service" pro odchozi UDP pakety
buffer=240 ;velikost bufferu
autoaddr=enable ;zmena odchozich RTP adres
anyssrc=disable ;prijmout prichozi SSRC
rtcp=enabled ;povolit RTCP
rtcp_interval=4500
drillhole=enable ;prochazeni pres NAT nebo firewall
thread=normal ;priorita zpracovani vlakna

ysigchan.conf

[general]
debuglevel=9,9,9,9,9
debuglevel_engine=9,9,9,9,9
enable=yes
[[link1] ;pro signalizaci SS7
enable=yes
type=ss7-isup
pointcodetype=ITU
pointcode=4
defaultpointcode=4
remotepointcode=3
lockgroup=yes
earlyacm=yes
sig=span1-1
voice=span1
strategy=increment
strategy-restrict=even
chansync=10000
numplan=isdn
netindicator=national
numtype=national
netind2pctype=ITU,ITU,ITU,ITU
presentation=allowed

screening=user-provided
format=alaw
print-messages=yes
extended-debug=yes
userparttest=0
inn=yes
location=U
emergency=no
[linkset3]
type=ss7-mtp3
netind2pctype=ITU
local=ITU,4
adjacent=ITU,3
autostart=yes
link=link1
link1.sig=span1-1
emergency=no
[link1]
type=ss7-mtp2
autostart=yes
emergency=no
fillink=yes
rxunderrun=0
[link2] ;pro signalizaci QSIG
type=isdn-pri-cpe
enable=yes
sig=span1-2
switchtype=qsig
strategy=increment
format=alaw
print-messages=yes

ysipchan.conf

[general]

port=5060
addr=192.168.10.174 ;bind adresa
registrar=enable ;pouziti jako registračního serveru

zapcard.conf

[general]
format=alaw
priority=normal ;priorita zpracovani
[span1-1] ;pro signalizaci SS7
type=E1 ;typ
offset=0 ;posun
voicechans=2-31 ;rozsah hlasovych kanalu
sigchan=1 ;signalizacni kanal
[span1-2] ;pro signalizaci Q-sig
type=E1 ;typ
offset=0 ;posun
voicechans=1-15.17-31 ;rozsah hlasovych kanalu
sigchan=16 ;signalizacni kanal
[span2] ;pro analog
offset=0
type=FXS
voicechans=34 ;analogovy kanal 1 emulovany jako 34.
dtmfdetect=disable ;detekce DTMF
[span3] ;pro analog
offset=0
type=FXS
voicechans=35 ;analogovy kanal 2 emulovany jako 35.
dtmfdetect=disable

Příloha 2 – Výpis debug zpráv pro testovací volání Z účtu IAX 102 na analog 104

```
-----  
IAX (0x06) - ACK (0x00000004)  
  Outgoing to 192.168.10.165:4569 (Local address: 192.168.10.174:4569)  
  Call (Local:Remote): 9551:14961. Timestamp: 3. Retrans: false. Sequence numbers: Out: 0 In: 1  
-----  
<iax/5:ALL> Created incoming [0x96572c8]  
<INFO> Could not classify call from '3004', wasted 5 usec  
<cdrbuild:INFO> Got message 'call.route' for untracked id 'iax/5'  
<INFO> Routing call to '104' in context 'default' via 'analog/ana-link2' in 246 usec  
<pbxassist:CALL> Created assistant for 'iax/5'  
<analog/2:CALL> Outgoing call on line ana-link2/35 caller=3004 called=104 [0x9649d80]  
<iaxengine:INFO> Sending frame.  
-----  
IAX (0x06) - ACCEPT (0x00000007)  
  Outgoing to 192.168.10.165:4569 (Local address: 192.168.10.174:4569)  
  Call (Local:Remote): 9551:14961. Timestamp: 3. Retrans: false. Sequence numbers: Out: 0 In: 1  
  FORMAT: 0x00000008 (G.711 a-law)  
  CAPABILITY: 0x00000008 (G.711 a-law)  
-----  
<pbxassist:NOTE> Channel 'iax/5' already assisted!  
<pbxassist:CALL> Created assistant for 'analog/2'  
<iaxengine:INFO> Received frame.  
-----  
IAX (0x06) - ACK (0x00000004)  
  Incoming from 192.168.10.165:4569 (Local address: 192.168.10.174:4569)  
  Call (Local:Remote): 9551:14961. Timestamp: 3. Retrans: false. Sequence numbers: Out: 1 In: 1
```

Z H.323 102 na SIP 710 (Barca) s pouzitim yrtpchan.conf

astStartResponse

<link1/Q931:INFO> Received message (0xb7025e10)

ALERTING

[From initiator=false CallRef=3]

Progress indicator

<sig/3:CALL> Call ringing [0xb701e638]

3:09.660 ThreadID=0xb6e79b70 h323.cxx(2792) H323 Answering call:

AnswerCallAlertWithMedia

3:09.660 ThreadID=0xb6e79b70 h323.cxx(4445) H245 Default

OnSelectLogicalChannels, FastStartResponse

<h323:INFO> External RTP address 192.168.10.43:49174

<yrtsp:INFO> RTP starting format 'alaw' payload 8 [0xb7021ed8]

<h323:INFO> External RTP address 192.168.10.43:49174

3:09.660 ThreadID=0xb6e79b70 h323.cxx(3376) H225 Accepting fastStart for 2

channels

3:09.660 ThreadID=0xb6e79b70 channels.cxx(920) H323RTP OnSendingPDU

3:09.660 ThreadID=0xb6e79b70 h323.cxx(2099) H225 Built fastStart for G.711-ALaw-

64k <1>

3:09.660 ThreadID=0xb6e79b70 channels.cxx(920) H323RTP OnSendingPDU

3:09.660 ThreadID=0xb6e79b70 h323.cxx(2099) H225 Built fastStart for G.711-ALaw-

64k <1>

3:09.661 ThreadID=0xb6e79b70 h323pdu.cxx(553) H225 Sending PDU

[ip\$192.168.10.174:1720/ip\$192.168.10.43:2503] : alerting

3:09.661 ThreadID=0xb6e79b70 h323.cxx(4374) H323

InternalEstablishedConnectionCheck: connectionState=AwaitingLocalAnswer

fastStartState=FastStartAcknowledged

3:09.660 ThreadID=0xb6e79b70 h323.cxx(2792) H323 Answering call:

AnswerCallAlertWithMedia

3:09.660 ThreadID=0xb6e79b70 h323.cxx(4445) H245 Default

OnSelectLogicalChannels, FastStartResponse

<h323:INFO> External RTP address 192.168.10.43:49174

<yrtsp:INFO> RTP starting format 'alaw' payload 8 [0xb7021ed8]

<h323:INFO> External RTP address 192.168.10.43:49174

3:09.660 ThreadID=0xb6e79b70 h323.cxx(3376) H225 Accepting fastStart for 2

channels

Z H.323 102 na analog 104

```
<pbxassist:CALL> Created assistant for 'h323/3'  
<yrtplib:INFO> Session 'yrtplib/1419192802' 0x88b3728 bound to 192.168.10.174:30306 +RTCP  
[0x88bf950]  
<h323/3:INFO> Got remote RTP address 192.168.10.43:49166 [0x88b4188]  
<h323/3:INFO> YateH323Connection::OnAnswerCall caller='491 [192.168.10.43]' chan=0x88b51e8  
[0x88b4188]  
<h323/3:INFO> GetRemotePartyNumber()=""  
<h323/3:INFO> Called number (alias) is '104'  
<INFO> Routing call to '104' in context 'default' via 'analog/ana-link2' in 340 usec  
<h323/3:NOTE> Formats changed to 'alaw'  
<analog/3:CALL> Outgoing call on line ana-link2/35 caller=(null) called=104 [0xb7619be0]  
<pbxassist:CALL> Created assistant for 'analog/3'  
<INFO> DataTranslator::attachChain [0x88c01a0] '(null)' -> [0xb7618c68] 'mulaw' not possible  
<analog/3:CALL> status=answered [0xb7619be0]  
<INFO> DataTranslator::attachChain [0xb7619e58] 'mulaw' -> [0x88bf5c0] '(null)' not possible  
<h323:INFO> External RTP address 192.168.10.43:49166  
<yrtplib:INFO> RTP starting format 'alaw' payload 8 [0x88bf950]  
<h323:INFO> External RTP address 192.168.10.43:49166  
<h323/3:INFO> YateH323Connection::OnEstablished() [0x88b4188]  
<yrtplib:WARN> Initial timeout in channel h323/3 wrapper [0x88bf950]  
<analog/3:CALL> status=hangup reason=normal [0xb7619be0]  
<analog/3:CALL> status=destroyed reason=normal [0xb7619be0]  
<pbxassist:INFO> Assistant for 'analog/3' hangup  
<pbxassist:INFO> Chan 'h323/3' disconnected in state 'new', reason 'normal'  
<pbxassist:INFO> Assistant for 'h323/3' disconnected, reason 'normal'  
<h323/3:INFO> YateH323Connection::stoppedExternal(receive) chan=(nil) [0x88b4188]  
<h323/3:INFO> YateH323Connection::stoppedExternal(send) chan=(nil) [0x88b4188]  
<pbxassist:INFO> Assistant for 'h323/3' hangup  
<h323/3:INFO> YateH323Connection::OnCleared() error: '(null)' reason: EndedByLocalUser (0)  
[0x88b4188]
```

Ze SIP 101 na SIP 710 (Barca)

< sip:INFO> Sending code 100 0x8992d20 to 192.168.10.43:5060

SIP/2.0 100 Trying
Via: SIP/2.0/UDP 192.168.10.43:5060;branch=z9hG4bK-d8754z-8c4ba83724e31e3d-1---d8754z-
;received=192.168.10.43;rport=5060
From: "101"< sip:101@192.168.10.174;transport=UDP>;tag=4c35c54d
To: < sip:710@192.168.10.174;transport=UDP>
Call-ID: YzMzYWUwM2M0NTA2NzUwZWJmMjc1YTEzNDg0ZTk4ZGQ.
CSeq: 1 INVITE
Server: YATE/3.1.0
Content-Length: 0

< sip/2:ALL> YateSIPConnection::YateSIPConnection(0x89982b0,0x89977d0) [0x8988888]
< pbxassist:CALL> Created assistant for 'sip/2'
< pbxassist:NOTE> Channel 'sip/2' already assisted!
< link1/B:NOTE> ZapCircuit(31). IOCTL(SetToneDetect) failed on channel 31 (param=3). 25:
Inappropriate ioctl for device [0xb76237c8]
< sig/6:CALL> Outgoing call from=101 to=710 trunk=link1 sigcall=0xb762f6b8 [0xb7635118]
< pbxassist:CALL> Created assistant for 'sig/6'
< sig/6:CALL> Call ringing [0xb7635118]
< NOTE> Choosing started 'audio' format 'gsm' [0x8992470]
< sip:INFO> Sending code 180 0x899dd40 to 192.168.10.43:5060

SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 192.168.10.43:5060;branch=z9hG4bK-d8754z-8c4ba83724e31e3d-1---d8754z-
;received=192.168.10.43;rport=5060
From: "101"< sip:101@192.168.10.174;transport=UDP>;tag=4c35c54d
To: < sip:710@192.168.10.174;transport=UDP>;tag=157501719
Call-ID: YzMzYWUwM2M0NTA2NzUwZWJmMjc1YTEzNDg0ZTk4ZGQ.
CSeq: 1 INVITE
Server: YATE/3.1.0
Contact: < sip:710@192.168.10.174:5060>
Allow: ACK, INVITE, BYE, CANCEL, REGISTER, REFER, OPTIONS, INFO
Content-Type: application/sdp
Content-Length: 186

Ze 710 (Barca) na analog 104

ZapCircuit(1). IOCTL(SetToneDetect) failed on channel 1 (param=3). 25: Inappropriate ioctl for device [0xb76237c8]

<sig/4:CALL> Incoming call from=710 to=104 trunk=link1 sigcall=0x899aed0 [0x8999398]

<pbxassist:CALL> Created assistant for 'sig/4'

<analog/2:CALL> Outgoing call on line ana-link2/35 caller=710 called=104 [0x899a5a8]

<pbxassist:NOTE> Channel 'sig/4' already assisted!

<pbxassist:CALL> Created assistant for 'analog/2'

<sig/4:CALL> Call ringing [0x8999398]

<sig/4:CALL> Call ringing [0x8999398]

<sig/4:CALL> Call ringing [0x8999398]

<sig/4:CALL> Call ringing [0x8999398]

<analog/2:CALL> status=answered [0x899a5a8]

<sig/4:CALL> Call answered [0x8999398]

<link1/B:NOTE> ZapCircuit(1). Unable to send unknown event 23 [0xb7624260]

<analog/2:CALL> status=hangup reason=normal [0x899a5a8]

<analog/2:CALL> status=destroyed reason=normal [0x899a5a8]

<sig/4:CALL> Call hangup. Reason: 'normal' [0x8999398]

<sig/4:CALL> Call destroyed. Reason: 'normal'. No signalling call [0x8999398]

SS7 počáteční inicializace

<INFO> Creating new message dispatching thread (0 running)

<link1/ISUP:INFO> Starting emergency proving interval [0xb7522f98]

<link1/ISUP:INFO> Proving period ended, link operational [0xb7522f98]

<linkset3:NOTE> Placing link 0 'link1/ISUP' in service [0xb7522c90]

<link1/ISUP:NOTE> Link inhibition changed 0x03 -> 0x02 [0xb7522f98]

<linkset3:NOTE> Linkset is not operational [0xb7522c90]

<ss7snm:NOTE> Link ITU,0-0-4:0 is up [0xb7522aa8]

<ss7snm:MILD> Could not changeback link 0, activating anyway [0xb7522aa8]

<link1/ISUP:NOTE> Link inhibition changed 0x02 -> 0x00 [0xb7522f98]

<linkset3:NOTE> Linkset is operational [0xb7522c90]

<ss7router:NOTE> Restart of SN initiated [0xb751b8e0]

<link1/ISUP:INFO> L3 (0xb751b8e4,'ss7router') is not operational sls=-1. Remote User Part is available

<link1/ISUP:INFO> L3 (0xb7522c90,'linkset3') is operational sls=0. Remote User Part is available

<sig:INFO> SigNotifier [0xe374c0] received a notify

<link1/ISUP:INFO> Sending message (0x9dcdf18)

RSC [cic=2 label=0-0-4:0-0-3:2]

<CpuModule:NOTE> Updating CPU core number from 1 to 2

<ss7router:NOTE> Restart of SN complete [0xb751b8e0]

SS7 ze 760 (Barca) na SIP 101

< sip:INFO > Sending code 501 0x86827c8 to 192.168.10.43:5060

SIP/2.0 501 Not Implemented

Via: SIP/2.0/UDP 192.168.10.43:5060;branch=z9hG4bK-d8754z-91931a5b685fe0fc-1---d8754z-;received=192.168.10.43;rport=5060

From: "101"< sip:101@192.168.10.174;transport=UDP >;tag=6c4be616

To: "101"< sip:101@192.168.10.174;transport=UDP >

Call-ID: MjJiZjJhNTUxMmMxOTQ1YjY4ZWU0ZTE1Y2RhMmVmN2U.

CSeq: 1 SUBSCRIBE

Server: YATE/3.1.0

Allow: ACK, INVITE, BYE, CANCEL, REGISTER, REFER, OPTIONS, INFO

Content-Length: 0

< yrtp:WARN > Initial timeout in channel sip/4 wrapper [0x8685d38]

< sip:INFO > Received 472 bytes SIP message from 192.168.10.43:5060

BYE sip:anonymous@192.168.10.174:5060 SIP/2.0

Via: SIP/2.0/UDP 192.168.10.43:5060;branch=z9hG4bK-d8754z-0d88a9a76fd9adce-1---d8754z-

Max-Forwards: 70

Contact: < sip:101@192.168.10.43:5060;rinstance=4f443414e9cb3f49;transport=UDP >

To: < sip:anonymous@192.168.10.174 >;tag=1142035981

From: < sip:101@192.168.10.43:5060;rinstance=4f443414e9cb3f49;transport=UDP >;tag=c714a464

Call-ID: 72473536@192.168.10.174

CSeq: 2 BYE

User-Agent: Zoiper rev.11137

Content-Length: 0

< sip:INFO > Sending code 100 0x8681918 to 192.168.10.43:5060

SIP/2.0 100 Trying

Via: SIP/2.0/UDP 192.168.10.43:5060;branch=z9hG4bK-d8754z-0d88a9a76fd9adce-1---d8754z-;received=192.168.10.43;rport=5060

From: < sip:101@192.168.10.43:5060;rinstance=4f443414e9cb3f49;transport=UDP >;tag=c714a464

To: < sip:anonymous@192.168.10.174 >;tag=1142035981

Call-ID: 72473536@192.168.10.174

CSeq: 2 BYE

Server: YATE/3.1.0

Content-Length: 0

< pbxassist:INFO > Assistant for 'sip/4' hangup

< pbxassist:INFO > Chan 'sig/5' disconnected in state 'new', reason '(null)'

< pbxassist:INFO > Assistant for 'sig/5' disconnected, reason '(null)'

< sig/5:CALL > Call hangup. Reason: 'normal-clearing' [0xb752a418]

< link1/ISUP:INFO > Sending message (0xb7525d78)

Příloha 3 – výpis použitelných funkcí v souboru regexroute.conf

\$() = a ; character
\$(\$) = a \$ character
\$(++N) = N+1
\$(--N) = N-1
\$(length, STRING) = length of STRING
\$(upper, STRING) = STRING converted to upper case
\$(lower, STRING) = STRING converted to lower case
\$(chr, N) = character with numeric code N
\$(hex, N[, LEN]) = little endian hexadecimal value of N with space between octets
\$(add, VAL1, VAL2[, LEN]) = VAL1+VAL2 left filled to LEN or length of VAL1
\$(sub, VAL1, VAL2[, LEN]) = VAL1-VAL2 left filled to LEN or length of VAL1
*\$(mul, VAL1, VAL2[, LEN]) = VAL1*VAL2 left filled to LEN or length of VAL1*
\$(div, VAL1, VAL2[, LEN]) = VAL1/VAL2 left filled to LEN or length of VAL1
\$(mod, VAL1, VAL2[, LEN]) = VAL1%VAL2 left filled to LEN or length of VAL1
\$(eq, VAL1, VAL2) = "true" if VAL1 = VAL2 (numerically), "false" otherwise
\$(ne, VAL1, VAL2) = "true" if VAL1 != VAL2 (numerically), "false" otherwise
\$(lt, VAL1, VAL2) = "true" if VAL1 < VAL2, "false" otherwise
\$(gt, VAL1, VAL2) = "true" if VAL1 > VAL2, "false" otherwise
\$(le, VAL1, VAL2) = "true" if VAL1 <= VAL2, "false" otherwise
\$(ge, VAL1, VAL2) = "true" if VAL1 >= VAL2, "false" otherwise
\$(streq, VAL1, VAL2) = "true" if VAL1 = VAL2 (string), "false" otherwise
\$(strne, VAL1, VAL2) = "true" if VAL1 != VAL2 (string), "false" otherwise
\$(random, STRING) = STRING with each ? character replaced with a random digit
\$(index, N, ITEM1, ITEM2, ...) = N-th (modulo length of list) item in list
\$(rotate, N, ITEM1, ITEM2, ...) = list rotated N (modulo length of list) times
\$(runid) = the current Engine run identifier
\$(nodename) = the node name the Engine runs as, may be empty
\$(threadname) = name of the thread that dispatched the message, may be empty
\$(dispatching) = the reentry depth, 0 if the message is not generated locally
\$(transcode, FLAGS, FORMAT1, FORMAT2, ...) = list of formats the input can be transcoded into
e - exclude initial formats form generated list
r - allow rate conversion (for use with wideband)
c - allow changing channels number
Note that functions ++, --, index and rotate will automatically update N if it is a variable in the \$varname format.

Příloha 4 – Možnosti parametrů pro spuštění Yate

Parametry:

-h, --help	zobrazí nápovědu a ukončí se
-V, --version	zobrazí verzi a ukončí se
-v	zvýší počet zobrazovaných informací pro ladění (možno použít vícekrát)
-q	sníží počet zobrazovaných informací pro ladění (možno použít vícekrát)
-d	daemonify, potlačí výstup bez přihlášení
-s	Supervised, restart při pádu nebo uzamčení
-r	povolí pokračování v zápisu do souboru (zapotřebí -s a -l)
--service	spustí jako službu Windows
--install	nainstaluje službu Windows
--remove	odstraní službu windows
-p filename	zapiše PID (project initiation document) do souboru
-l filename	Log do souboru
-n configname	použije uvedený konfigurační soubor ()
-c pathname	cesta k adresáři s konfiguračními soubory (/etc/yate)
-m pathname	cesta k adresáři s moduly (/usr/lib/yate/modules)
-w directory	změní pracovní adresář
-C	Enable core dumps if possible
-D [options]	Special debugging options
a	Abort if bugs are encountered
m	Attempt to debug mutex deadlocks
l	Try to keep module symbols local
c	Call dlclose() until it gets an error
u	Do not unload modules on exit, just finalize
i	Reinitialize after 1st initialization
x	Exit immediately after initialization
w	Delay creation of 1st worker thread
o	Colorize output using ANSI codes
s	Abort on bugs even during shutdown
t	Timestamp debugging messages relative to program start
e	Timestamp debugging messages based on EPOCH (1-1-1970 GMT)
f	Timestamp debugging in GMT format YYYYMMDDhhmmss.uuuuuu