SIGNALIZACE SIP

Ing. Miroslav VOZŇÁK, Ph.D.
VŠB – TUO, FEI, Katedra elektroniky a telekomunikační techniky,

Abstrakt: *Session Initiation Protocol (SIP) je signalizační protokol, který je užíván k řízení multimediálních relací v IP sítích, obsahuje zprávy pro sestavení, udržení a ukončení spojení. Jádro protokolu je definováno v RFC 3261, jeho oblíbenosti prospělo textově-orientované kódování, poměrně snadná rozšiřitelnost a dynamický vývoj charakteristický pro normy vznikající v rámci IETF. Příspěvek je zaměřen na popis prvků VoIP řešení s protokolem SIP, popis zpráv (metody a odpovědi) s doplněním o příklady autentizace při registraci, rozdílů chování na stavové a bezestavové SIP Proxy. Přednáška bude obsahovat praktickou ukázku spojení s rozklíčováním zachycené komunikace a s vysvětlením obsahu hlaviček jednotlivých zpráv. Rovněž budou vysvětlena i důležitá rozšíření pro SIP a jejich využití (Presence, Instant Messaging). Závěr příspěvku bude patřit otevřeným řešením s podporou SIP protokolu, zejména SIP Express Router a Asterisk, které jsou využívány v akademickém prostředí.*

1 Úvod

IP telefonie je jednou z nejvýznamnějších změn v dějinách telefonie. V první etapě prosazování IP telefonie se používal nejčastěji termín konverze hlasových a datových sítí, tak dnes už nehovoříme o konverzi, ale rovnou o změně anebo náhradě. Veřejná telefonní síť se téměř 100 let vyvíjela na analogovém principu, koncové telefonní přístroje i ústředny byly analogové a také všechny přenosy byly prováděny analogovou formou. Od poloviny minulého století byly učiněny první kroky k digitalizaci a standardizace PCM (pulzní kódové modulace, ITU-T G.711) v roce 1972 změnila oblast telekomunikací a otevřela brány digitálního světa, viz. [1], [2]. V něm je charakteristický kanál 64 kbit/s a spojovací pole propojovacích prvků je založeno na propojování toků 64 kbit/s. Dnes jsme v další etapě vývoje telefonie, telefonování je aplikací v sítích s Internet protokolem.

Za počátek IP telefonie je považován únor 1995, kdy izraelská společnost Vocaltec přišla s komerčním produktem Internet Phone. Nové techniky zpracování hlasu, především kodeky s lineární predikcí, umožnily redukovat hovorové pásmo při zachování kvality a jejich nasazení se uplatnilo právě v IP telefonii. Dominantní se stalo kódování dle ITU- T G.729 a G.723.1., přičemž povinné pro všechny zařízení je schopnost dorozumět se na ITU- T G.711. Využití Internetu jako páteřní infra-

struktury pro telefonii snižuje náklady na telefonování, s rozšiřováním IP telefonie klesají provozní režie a zvyšuje se konkurenceschopnost služby, což stimuluje její rozšiřování a snižuje ceny, jedná se tak o efekt spirály. Jsou zde ovšem nové otázky bezpečnosti (zneužití VoIP systémů, podvržení cizí identity, spam, zajištění utajení a integrity hovoru). Výhody, které přináší IP telefonie vidíme ve dvou rovinách:

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

Pokud jsme si uvedli výhody, musíme říci i nevýhody, které opět můžeme rozdělit do dvou oblastí:

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

Voice over IP prochází evolucí stejně jako jakákoliv rozvíjející se technologie. IP telefonie se v počátcích orientovala striktně na oblast Internetu, kdy se používaly samostatné IP telefony především ve formě aplikací pro PC a nebylo realizováno propojení s PSTN, jednalo se o pionýrské začátky IP telefonie. Dnes je nabízena IP telefonie jako plnohodnotná náhrada telefonních přípojek a klasických pobočkových ústředen.

V sítích s protokolem IP se hlas přenáší v paketech RTP (Real Time Protocol), které na transportní vrstvě používají protokol UDP. Formát tohoto paketu je dán doporučením IETF z roku 1996 s označením RFC1889/1890 pro RTP/RTCP, přičemž RTP řeší vlastní přenos hlasové informace a RTCP kontrolní mechanizmus v doručování RTP (Real Time Control Protocol). Novější implementace používají RFC3550 z roku 2003 nebo ještě novější SRTP (Secure RTP) dle RFC3711 z roku 2004. Během přenosu RTP dostáváme informace o počtu ztracených paketů a proměnném zpoždění pomocí kontrolního protokolu RTCP. Jeho zajímavým rozšířením je RTCP XR (Control Protocol Extended Reports), který definuje soubor metrik pro hodnocení VoIP kvality volání, určování problémů, viz. [3], [4]. RTCP XR

jednak umožňuje zobrazení hodnoty MOS či R-faktoru přímo na koncových zařízeních, ale mnohem užitečnější je sledování těchto kvalitativních parametrů a vyhodnocení aktuální dostupné kvality pro konkrétní destinace, což může vést například ke změně směrování volání v síti (používá se kupříkladu v síti AARNET).

Z pohledu signalizačních protokolů máme následující možnosti. Nejstarším doporučením je ITU-T H.323 pro multimediální komunikaci v paketových sítích, H.323 zastřešuje řadu standardů, první verze je z roku 1996, poslední verze H.323v6 je z roku 2006. Pracovní skupina ITU-T SG16, která odvedla podstatnou část práce na H.323 se dnes zabývá tvorbou nového standardu ITU-T H.325, což svědčí i o tom, že o osudu dalšího vývoje H.323 je rozhodnuto, viz. [5]. Vyvíjený H.325 bude zcela novým standardem pro IP telefonii a jeho uvolnění je naplánováno na rok 2009. Dalším protokolem je SIP, tomu se budeme věnovat ve všech dalších kapitolách. Poprvé byl SIP uvolněn v roce 1999 a dnes je dostupný ve dvou verzích RFC, k SIPu se váže ale dalších zhruba sedmdesát RFC, které rozšiřují jeho možnosti. Ve veřejných sítích se používá softswitch architektura, která vyžaduje oddělit média, řídící a signalizační funkce a k tomu účelu byl vyvinut master-slave protokol MGCP (Media Gateway Control Protocol) a Megaco. MGCP byl poprvé uvolněn IETF v roce 1999. Další vývoj pokračoval společně s ITU-T pod označením Megaco/H.248 specifikovaným o rok později než MGCP (H.248 a Megaco jsou ekvivalentní jména pro stejný protokol). Rok byla ovšem dostatečně dlouhá doba k tomu, aby někteří výrobci implementovali MGCP a tím umožnili jeho další vývoj. Veškerá RFC pro MGCP jsou jako informativní, zatímco pro Megaco mají statut standardu. Řada lidí v oboru používá stejné označení pro MGCP a Megaco/H.248 či navzájem je zaměňuje, což je chyba, jelikož jsou to dva různé protokoly, ačkoliv v mnoha částech podobné.

2 Vlastnosti SIPu

SIP byl vyvíjen od roku 1996 pracovní skupinou MMUSIC (Multiparty Multimedia Session Control) v rámci IETF (Internet Engineering Task Force). V roce 1999 byl předložen ve formě navrhovaného standardu (Proposed Standard) v RFC 2543. Téhož roku na popud IETF vznikla nová pracovní skupina, nazvaná příznačně SIP, která převzala vývoj hlavního jádra protokolu. Její práce v květnu roku 2002 vyústila v nový standard RFC 3261. SIP je signalizační protokol pracující na aplikační vrstvě, tento protokol byl navržen tak, aby byl snadno implementovatelný, rozšiřitelný a dostatečně flexibilní. Specifikace je dostupná ve formě několika doporučení RFC, nejdůležitější je RFC3261 jež obsahuje jádro protokolu. Protokol je užíván pro sestavení, modifikaci a ukončení spojení s jed-

ním nebo více účastníky. SIP není jediný protokol, který je potřebný pro komunikující zařízení. Ve spojení se SIPem jsou nejčastěji používány ještě dva další protokoly, RTP a SDP. RTP protokol je užíván k přenosu multimédií v reálném čase (real-time), tento protokol umožňuje přenášet hlas nebo video v paketech pomocí IP. Dalším důležitým protokolem je SDP, který je užíván k popisu vlastností účastníků spojení. Tento popis je pak použit k vyjednání parametrů spojení všech zařízení účastnících se spojení (vyjednání kodeků transportního protokolu), uvažuje se však o nahrazení novým protokolem založeným na XML.

SIP byl navržen v souladu s modelem Internetu. Jde tedy o end-to-end orientovaný signalizační protokol, což znamená, že veškerá logika je uložená v koncových zařízeních (vyjma směrování SIP zpráv), koncové zařízení zná i jednotlivé stavy komunikace, tím je zvýšena odolnost komunikace proti chybám. Cena, která se musí zaplatit za decentralizaci a dostupnost služby, je vyšší režie v hlavičkách zpráv (zprávy jsou posílány end-to-end). Nepochybně stojí za zmínku, že end-to-end koncept SIPu je významná odlišnost od klasického řešení PSTN (Public Switched Telephone Network), kde logika je uložena v síti a koncová zařízení jsou podstatně jednodušší. Cílem SIPu je zajistit stejnou funkcionalitu jakou mají klasické PSTN, ale end-to-end návrh umožní SIP sítím vyšší výkonnost a otevře implementaci nových služeb, které mohou být jen ztěží nasazeny v klasických PSTN. Komunikace v SIPu probíhá výměnou dvou typů zpráv, požadavků a odpovědí. Klient i server jsou logickou částí jednoho prvku.

SIP je založen na HTTP protokolu. HTTP protokol má formát hlaviček zpráv dle RFC822. HTTP je nepochybně nejúspěšnějším a nejpoužívanějším protokolem v Internetu. HTTP může být taky chápán jako signalizační protokol, protože UA (user agents) jej používají, aby sdělili HTTP serveru, o který dokument se zajímají. SIP je užíván k přenosu popisu parametrů relace, popis je zakódován dovnitř dokumentů používajících SDP. Oba protokoly (HTTP a SIP) zdědili kódování hlaviček zpráv z RFC822. Volba osvědčeného formátu zaručuje robustnost a nadčasovost, viz. [6], [7].

3 Adresace

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

;*sip:user@host:porturi-parameters*

Jak můžeme vidět, SIP URI se skládá z části user a z části host , obě části jsou oddělené znakem @. SIP URI je podobná e-mailové adrese, je doporučeno používat stejnou adresu pro e-mail i SIP, takže URI může být snadné si zapamatovat. Část user identifikuje uživatele v doméně prezentované v části host, která může být zadána pomocí jména nebo IP adresy. Pokud není uvedeno číslo portu, tak se předpokládá použití všeobecně známého portu 5060.Parametry mohou nést další volitelné informace. Doménová část URI je adresována s využitím DNS, což dává adresaci vysokou flexibilitu. V následující tabulce 1 jsou uvedeny příklady URI.

URI	použití adresy	doporučení
sip: nebo sips:	SIP a Secure SIP adresa	RFC 3261
tel:	Telefonní čísla	RFC 3999
pres:	Prezence	RFC 3861
im:	Instant Message	RFC 3861
http:	Web	RFC 2616
h323	H.323 URL	RFC 3508

Tab. 1: URI a odkazy na doporučení

4 Prvky SIP řešení

Ačkoliv v nejjednodušší konfiguraci je možné použít dva UA posílající si navzájem SIP zprávy, typická SIP síť bude obsahovat více než jeden typ prvků. Základními SIP prvky jsou:

- user agents,
- proxies, registrars, and redirect servers.

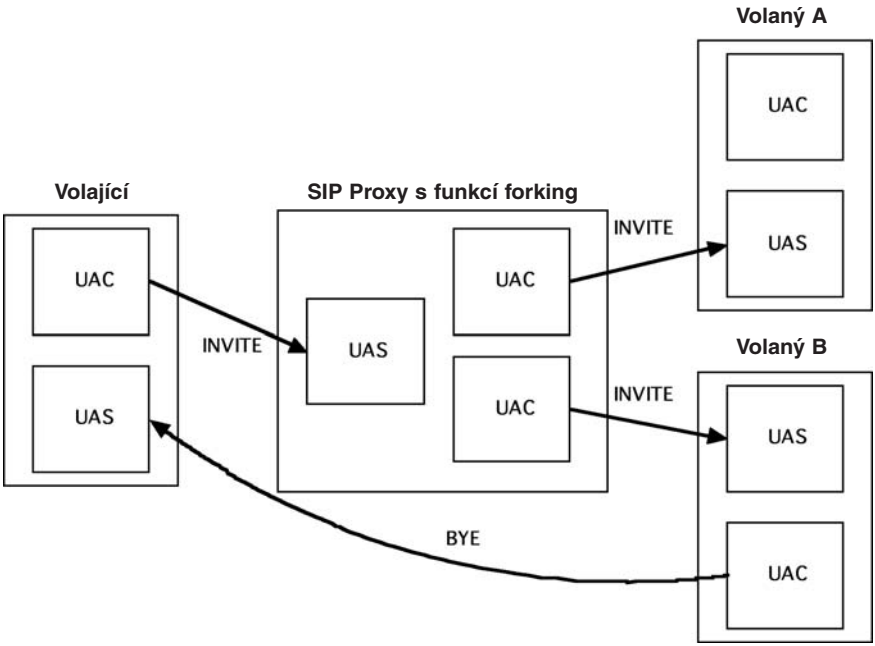
Jednotlivé servery jsou většinou prezentovány jako logické části SIP serveru, jelikož je často efektivní je provozovat společně na jednom HW. Koncové body sítě Internet, které užívají SIP ke vzájemnému spojení jsou nazývány jako user agents (UA). UA obvykle jsou představovány koncovými terminály ve formě HW SIP telefonu nebo aplikace, SIP UA mohou být i mobilní telefony, PSTN brány (GW), PDA, IVR systémy atd.

UA jsou vztaženi k částem User Agent Server (UAS) a User Agent Client (UAC). UAS a UAC jsou pouze logické entity, každý UA obsahuje UAC a UAS:

- UAC je část vysílající požadavky a přijímající odpovědi,
- UAS je část přijímající požadavky a odesílající odpovědi.

Požadavek a odpověď jsou dva základní typy SIP zpráv. Protože koncové zařízení téměř vždy obsahuje UAC a UAS, tak používáme pouze označení UA namísto UAC a UAS. Například, volající UA funguje jako UAC, když odesílá zprávu INVITE (požadavek na sestavení spojení) a přijímá odpověď na požadavek. Volaný UA se chová jako UAS, když obdrží zprávu INVITE a odesílá odpověď. Ale tato situace se mění, když volaný se rozhodne zavěsit a odesílá se zpráva BYE a ukončuje spojení. V tomto případě se volající chová jako UAC a volaný jako UAS.

Na obrázku 1 jsou tři UA a SIP proxy. Každý UA obsahuje UAC a UAS. Část SIP proxy, která přijímá INVITE od volajícího, funguje jako UAS. V případě větvení SIP proxy přeposílá požadavek na dvě místa zároveň a vytváří dva UAC, každý odpovídající jednomu větvení na obrázku. V našem příkladě volaný UAS B vyzvedl a později zavěsil, tím vyslal požadavek BYE a zachoval se jako UAC.



Obr. 1: Interakce mezi UAC a UAS

4.1 SIP server

SIP umožňuje vytvořit infrastrukturu sítě hostitelů nazývaných jako proxy servers. Koncové terminály UA mohou odesílat zprávy na proxy server. Proxy servery jsou důležité entity v SIP infrastruktuře, zajišťující směrování žádostí o spojení dle aktuálního umístění adresáta, autentizaci, účtování a spoustu dalších důležitých funkcí. Nejdůležitější úloha proxy serveru je směrovat žádosti o sestavení spojení blíž k volanému. Při inicializaci sestavení spojení bude obvykle prohledávat řadu proxy serverů, dokud nenajde nějaký, který zná aktuální umístění volaného. Takže proxy bude přesměrovávat žádost o spojení přímo k volanému a volaný akceptuje nebo odmítne žádost o spojení. Máme dva základní typy SIP proxy serverů:

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

4.2 Stavová a bezestavová SIP Proxy

Stateless Servery servery jsou poměrně jednoduché a pouze přeposílají zprávy nezávisle na jejich vzájemné vazby. Zprávy jsou většinou v pořádku z hlediska souslednosti a významu v signalizaci, stateless proxy servery neumí kontrolovat jejich výměnu z hlediska smysluplnosti, takže může vyjímečně docházet k nekorektním stavům, které musí být ošetřeny na úrovni koncového zařízení. Stateless proxy servery jsou jednoduché, ale rychlejší než stateful proxy servery. Využití mají např. pro snížení zátěže, pro jednoduché překládání zpráv a směrování. Jedna z nevýhod stateless proxy serverů je, že nejsou schopné zachytit opakování zpráv a provádět dokonalejší směrování, např. větvení nebo předání. Ze zachycené signalizace poznáme, že se jedná o stateless server, pokud budeme sledovat, zda zprávy pouze přeposílá.

Stateful Servery jsou Proxy servery s informací o stavech jsou mnohem komplexnější. Po přijetí požadavku, server si vytvoří záznam stavu a tento stav drží, dokud nedojde k ukončení transakce. Některé transkace, zejména vytvořené zprávou INVITE, mohou trvat poměrně dlouho (dokud volaný nevyzvedne nebo se neukončí volání). Protože proxy servery s informací o stavech musí udržovat stav po celou dobu dané transakce, je jejich výkon limitován.

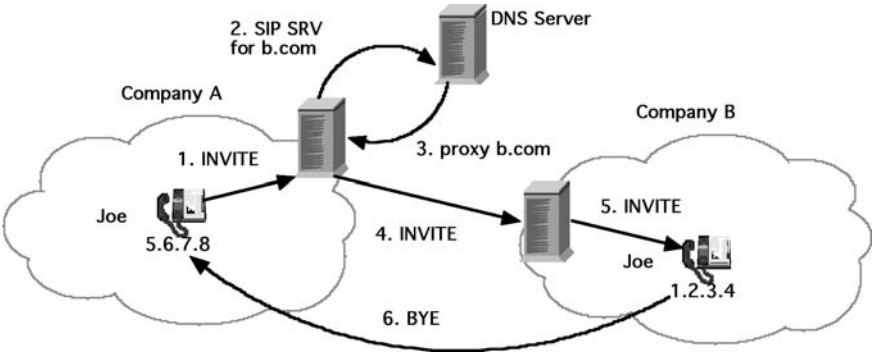
Schopnost přiřazovat SIP zprávy do transakcí dává serveru některé zajímavé vlastnosti, například může provádět větvení, na přijetí jedné zprávy, může být odesláno více zpráv. Stateful proxy server může zachytit opakování zpráv, protože ze

stavu transakce lze určit, jestli byla stejná zpráva už přijata (stateless proxy nemůže ověřovat, protože nedrží stav). Proxy server s informací o stavech může provádět komplikovanější metody nalezení uživatele, například možnost zkusit volat na telefon v zaměstnání a v případě neohlášení volání přesměrovat na mobilní telefon. Bezestavová proxy to nemůže, protože nemá informaci, zda bylo dosaženo cíle či nikoliv. Většina SIP proxy serverů je s informací o stavech, často nabízejí účtování, větvení, některé druhy podporují i NAT. Ze zachycené signalizace poznáme stateless SIP proxy dle informativních odpovědí, tato proxy nejdříve na žádost odpoví a až potom ji přepošle dál. SIP Proxy se dále dělí na:

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

5 Typický scénář spojení

Typická konfigurace je taková, že každá jednotka (např.firma) má vlastní SIP server, který je užíván všemi UA v rámci administrované jednotky, v drtivé většině případů se jedná o jednu doménu. Může se jednat ovšem i o případ, že jeden SIP server obsluhuje více domén, potom jde o multidoménovou SIP Proxy. Multidoménová Proxy je sice umístěná v konkrétní doméně, ale pomocí pole *realm* umí určit, do které domény uživatel patří. Předpokládejme, že jsou dvě firmy A a B a každá z nich má svůj Proxy server ve své doméně. Obrázek ukazuje jak zaměstnanec Joe ve firmě A inicializuje spojení se zaměstnancem Bobem z firmy B.



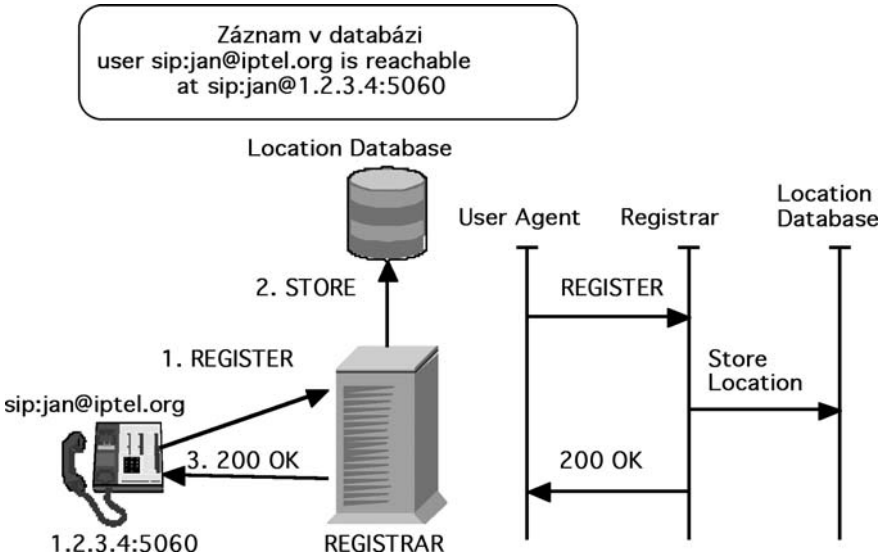
Obr. 2: Inicializace spojení Joe – Bob

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

Zmínili jsme se o tom, že SIP proxy na *proxy.b.com* zná současnou polohu Boba, ale neřekli jsme, jak Proxy může lokalizovat uživatele. Bobův UA (SIP telefon) musí být registrován na *registrar serveru*. Registrar server je speciální část SIP serveru, která přijímá od uživatelů požadavky na registraci, tím získává informaci o jejich aktuální poloze (IP adresa, port a uživatelské jméno) a ukládá informaci do lokalizační databáze (location database). V lokalizační databázi v našem případě došlo k namapování adresy *sip:bob@b.com* k adrese *sip:bob@1.2.3.4:5060*. Lokalizační databáze je pak užívána Proxy serverem. Když Proxy server obdrží žádost pro *sip:bob@b.com*, vyhledá v lokalizační databázi záznam *sip:bob@1.2.3.4:5060* a tam pošle žádost. Registrar server je velmi často pouze jako logická část SIP serveru, jelikož je těsně svázán s Proxy serverem.

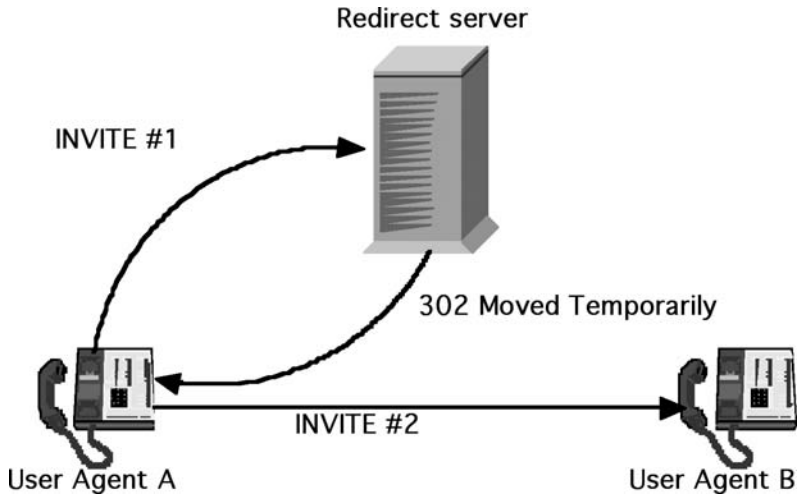
5.1 Redirect a Registrar server

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



Obr. 3: Registrar server

Každá registrace má limitovanou dobu platnosti, doba platnosti je v hlavičce kontaktu, do té doby musí UA obnovit registraci, jinak bude nedostupný.



Obr. 4: Redirect server a přesměrování volání

Entita, která přijímá požadavek a odesílá zpět odpověď obsahující lokalizaci konkrétního uživatele se nazývá *Redirect server*. Redirect server přijímá požadavky a vyhledává zamýšlené příjemce v lokalizační databázi vytvořené registrar serverem. Následně vytváří seznam aktuálních lokalizací uživatele a posílá jej odesílateli požadavku v odpovědi zařazené do třídy 3xx. Odesílatel požadavku poté dostává seznam destinací a odesílá další požadavky přímo na ně. Obrázek 4 ukazuje typické přesměrování.

6 Metody a odpovědi

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

- žádost (metoda),
- odpověď.

Žádosti jsou obvykle užívány k inicializaci procedury (sestavení, ukončení spojení) nebo oznamují příjemci požadavek na něco. Odpovědi jsou užívány k potvrzení, že žádost byla přijata a zpracována a obsahuje stav zpracování. Typická SIP žádost vypadá následovně:

INVITE sip:7170@iptel.org SIP/2.0
Via: SIP/2.0/UDP 195.37.77.100:5060
Max-Forwards: 10
;From: „jiri“ <sip:jiri@iptel.org>tag=76ff7a07-c091-4192-84a0-d56e91fe104f
To: <sip:jiri@bat.iptel.org>
Call-ID: d10815e0-bf17-4afa-8412-d9130a793d96@213.20.128.35
CSeq: 2 INVITE
Contact: <sip:213.20.128.35:9315>
User-Agent: Windows RTC/1.0
Proxy-Authorization: Digest username=„jiri“, realm=„iptel.org“,
algorithm=„MD5“, uri=„sip:jiri@bat.iptel.org“,
nonce=„3cef753900000001771328f5ae1b8b7f0d742da1feb5753c“,
response=„53fe98db10e1074
b03b3e06438bda70f“

```
Content-Type: application/sdp
Content-Length: 451

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

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

SIP žádost obsahuje v hlavičce jedno nebo více polí *Via*, jež jsou užívány k záznamu cesty žádosti. Následně jsou užívány ke směrování SIP odpovědí přesně takovou cestou, jakou byly odeslány. Naše INVITE zpráva obsahuje jedno pole *Via*, které bylo vytvořeno UA, který odeslal žádost. Z pole *Via* můžeme říct, že UA používá IP adresu 195.37.77.100 a port 5060.

Pole hlavičky *From* a *To* identifikuje iniciátora volání (volající) a příjemce (volaného). Pole *From* obsahuje parametr *tag*, který slouží jako identifikátor dialogu a bude popsán v další kapitole. Pole hlavičky *Call-ID* je identifikátor dialogu a jeho cílem je identifikovat zprávy náležející jednomu volání. Takovéto zprávy mají stejný identifikátor *Call-ID*. Ke správě pořadí požadavků je užíváno pole *CSeq*. Protože žádosti mohou být odeslány nespolehlivým přenosem, který může

způsobit zpřeházení zpráv, pořadové číslo se musí ve zprávě nacházet, aby příjemce mohl rozpoznat opakování přenosu a selektovat žádosti. V hlavičce je pole *Contact* obsahující IP adresu a port, na kterém odesílatel očekává další žádosti odesílané volaným.

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

6.1 Žádosti neboli metody

Žádosti neboli metody specifikované v RFC 3261 jsou následující:

- *INVITE je* žádost o inicializaci spojení nebo změnu parametrů již probíhajícího spojení,
- *ACK* tato zpráva potvrzuje přijetí odpovědi na žádost INVITE. Sestavení relace používá „3-way hand-shaking“, volaný periodicky opakuje odpověď (OK), dokud nepřijme ACK, což indikuje, že volající je stále připraven komunikovat,
- *BYE je* zpráva je užívána k ukončení spojení některou z komunikujících stran,
- *CANCEL* je užíván ke zrušení sestavovaného spojení, když volaný ještě nepotvrdil žádost INVITE a volající chce zrušit sestavování spojení,
- *REGISTER* smyslem žádosti je sdělit aktuální polohu uživatele. V této zprávě je přenášena informace o aktuální IP adrese a portu, na kterém může být uživatel zastižen. Registrace jsou časově limitovány a potřebují periodicky obnovovat,
- *OPTIONS* je žádost o zaslání schopností (vlastností) serveru nebo UA.

V tabulce 2 je uveden seznam SIP metod, význam a odkaz na RFC, jedná se o základní typy metod.

smysl žádosti	**název metody**	**doporučení**
sestavení relace	INVITE	RFC 3261
potvrzení na INVITE	ACK	RFC 3261
získání schopností entity	OPTIONS	RFC 3261
zrušení dosud nevyřízené žádosti	CANCEL	RFC 3261
ukončení existující relace	BYE	RFC 3261
registrace (svázáno s URI)	REGISTER	RFC 3261
přihlášení k odběru informací	SUBSCRIBE	RFC 3265
doručení informace (přihlášeným k odběru informací)	NOTIFY	RFC 3265
aktualizace stavu informace na server	PUBLISH	RFC 3903
požadavek jiného UA k relaci (např. inicializace spojení přes web)	REFER	RFC 3515
přenos zpráv Instant Message	MESSAGE	RFC 3428
aktualizace informace o stabu relace	UPDATE	RFC 3311
potvrzení prozatimní (1xx) odpovědi	PRACK	RFC 3262
přenos signalizačních informací během relace	INFO	RFC 2976

Tab. 2: URI a odkazy na doporučení

6.2 Odpovědi

Jestliže UA nebo Proxy server obdrží žádost, tak odesílá odpověď. Každá žádost musí být zodpovězena kromě ACK žádosti. Typická odpověď vypadá následovně:

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

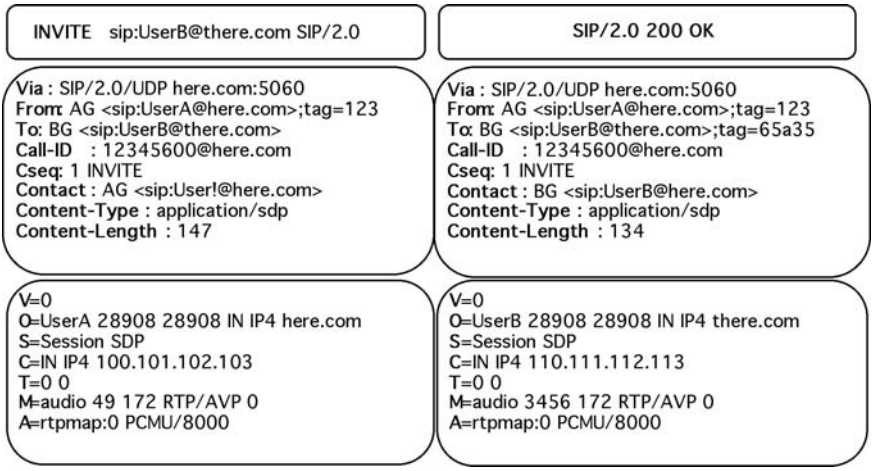
Jak můžeme vidět, odpovědi jsou velmi podobní žádostem, kromě prvního řádku. První řádek odpovědi obsahuje verzi protokolu (SIP/2.0), kód odpovědi (*reply code)*. Kód odpovědi je celé číslo z rozsahu 100 až 699 a označuje typ odpovědi. Celkem je 6 tříd odpovědí:

- ***1xx*** jsou informativní odpovědi, které jsou odesílány na žádosti, které byly přijaty, ale výsledek zpracování ještě není znám, na základě této odpovědi musí odesílatel zastavit opakování odesílání dané žádosti. Obvykle proxy servery odesílají odpovědi s kódem 100 (Trying), jestliže začínají zpracovávat INVITE a UA odesílají odpovědi s kódem 180 (Ringing), které oznamují vyzvánění volaného,
- ***2xx*** jsou pozitivní finální odpovědi, je to poslední odpověď, kterou odesílatel na svou žádost dostává, vyjadřuje výsledek zpracování konkrétní žádosti. Odpovědi s kódy 200 až 299 oznamují, že požadavek byl akceptován a úspěšně zpracován, například odpověď 200 OK je vyslána, jestliže uživatel akceptuje žádost INVITE. V případě větvení zprávy INVITE můžeme dosáhnout několik UAS a každý z nich bude akceptovat žádost. V tomto případě je každá odpověď rozlišena parametrem *tag* v poli *To*. Každá odpověď probíhá v odlišném dialogu s jedinečným identifikátorem dialogu,
- ***3xx*** odpovědi jsou užívány k přesměrování. Tyto odpovědi dávají informaci o nové poloze uživatele nebo alternativní službě, která má být použita. Pokud Proxy přijme žádost a nezpracuje ji z nějakého důvodu, tak vyšle volajícímu v odpovědi požadavek na přesměrování a vloží do odpovědi jiné umístění, které má být kontaktováno. Může to být jiná Proxy nebo aktuální umístění volajícího (z lokalizační databáze vytvořené registrar serverem). Volající následně znovu vyšle žádost na nové umístění, odpovědi 3xx jsou konečné,
- ***4xx*** jsou negativní konečné odpovědi a znamenají problém na straně odesílatele. Žádost nemohla být zpracována, protože obsahuje chybnou syntaxi,
- ***5xx*** znamenají problém na straně serveru. Žádost je zřejmě v pořádku, ale server selhal při zpracování, klient by měl obvykle požadavek zkusit znovu,
- ***6xx*** tento kód je vysílán, pokud žádost nemůže být splněna na žádném serveru, to je odpověď obvykle vysílaná serverem, když má informaci o konkrétním uživateli, např. UA vysílá *603 Decline response*, když odmítá žádost o sestavení spojení,

třída	**kód**	**typické příklady**
informativní a prozatimní	1xx	100 Trying 180 Ringing 183 Session Progress
úspěšné	2xx	200 OK 202 Accepted
přesměrování	3xx	300 Moved 302 Multiple Choices 305 Use Proxy
chyba u klienta	4xx	401 Unauthorized 403 Forbidden 415 Unsupported Media Type 486 Busy Here 428 Use Identity Header
chyba na serveru	5xx	501 Not Implemented 503 Service Unavailable
gobální chyba	6xx	600 Busy Everywhere 603 Decline

Tab. 3: Typické odpovědi

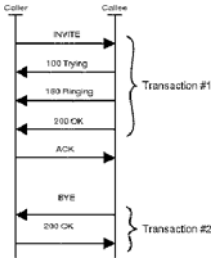
Kromě odpovědi odpovídající třídy, první řádka obsahuje popis *reason phrase*, obsažený kód je určen ke zpracování, což je snadno analyzovatelné a popis lidsky oznamuje výsledek., UA může popis zobrazit uživateli. Přiřazení žádosti k odpovědi je na základě pole *CSeq* v hlavičce, kromě pořadového čísla obsahuje také metodu korespondující žádosti, v našem případě to byla žádost REGISTER. Na obrázku 5 jsou pole žádosti INVITE a konečné odpovědi na ni 200 OK, zobrazeny jsou i obsahy SDP (popis médií).



Obr. 5: Žádost a odpověď

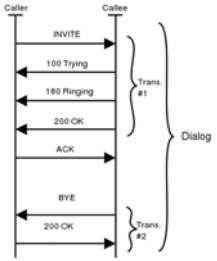
Následně si vysvětlíme, co je to *SIP transakce.* Ačkoliv bylo řečeno, že SIP zprávy jsou posílány sítí nezávisle, tak obvykle jsou uspořádány do transakcí agenty UA a určitými typy Proxy serverů. Transakce je sekvence SIP zpráv vyměňovaných mezi síťovými SIP prvky. Transakce obsahuje jednu žádost a všechny odpovědi vztažené k této žádosti. To může znamenat žádnou, jednu nebo i více prozatimních odpovědí a jednu nebo více konečných odpovědí (např. při větvení v Proxy na zprávu INVITE). Jestliže byla transakce zahájena žádostí INVITE , pak stejná transakce obsahuje i zprávu ACK, ale pouze tehdy, jestliže finální odpověď nebyla třídy 2xx, potom ACK není součástí dané transakce, důvodem je, že odpověď je zpráva 200 OK. Především UA jsou zodpovědni za opakování odpovědi 200 OK, dokud nepřijmou ACK.

SIP entity, které sledují tyto transakce, se nazývají *stateful*, tzn. se záznamem o stavech. Vytvořený stav je spojován s transakcí a je držen v paměti po celou dobu trvání transakce. Pokud přichází žádost nebo odpověď, tak je zkoušeno vždy přiřazení zprávy do probíhající transakce. Aby tato operace byla proveditelná, tak musí ze zprávy přečíst jednoznačný identifikátor transakce a porovnat jej s identifikátory probíhajících transakcí. Jestliže takováto transakce existuje, tak dochází k jejímu doplnění o další informace. V předchozím SIP RFC2543 byl identifikátor transakce odvozován ze všech důležitých informací v hlavičce zprávy (zahrnující pole *To, From, Request-URI a CSeq*), což bylo komplikované a zpomalovalo zpracování a při testech interoperability bylo zdrojem problémů.



Obr. 6: Transakce

V novém RFC3261 byl způsob výpočtu identifikátoru transakce zásadně změněn. Namísto komplikovaného určování z polí hlavičky teď obsahuje SIP zpráva identifikátor přímo v políčku *Via*. To je významné zjednodušení, ale stále existují staré implementace, které nepodporují nový způsob určování identifikátoru transakce, proto musí být určování zpětně kompatibilní s oběmi verzemi. Obrázek 6. znázorňuje zprávy zařazené do transakcí během konverzace dvou UA.



Obr. 7: Dialog

V předchozím příkladě jsme si ukázali dvě transakce, jedna transakce obsahovala zprávu INVITE a odpověď, druhá obsahovala zprávu BYE a následnou odpověď. Obě transakce však spolu souvisí a náleží do jednoho *dialogu*.

Dialog bychom mohli definovat jako soubor SIP zpráv peer-to-peer mezi dvěma UA, které mají vzájemnou spojitost. Dialogy popisují řazení a směrování zpráv mezi dvěma koncovými body. Dialogy jsou identifikované pomocí pole *Call-ID*, *From* a *To*. Zprávy, které mají tyto tři identifikátory stejné, tak náleží jednomu dialogu. Ukázali jsme si, že pole *CSeq* je užíváno k řazení zpráv, je používáno k řazení zpráv uvnitř dialogu. *CSeq* číslo určuje transakci uvnitř dialogu, protože jsme

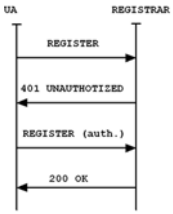
řekli, že žádost a přidružená odpověď se nazývá transakce. To znamená, že v každém směru může být uvnitř dialogu aktivní pouze jedna transakce. Mohli bychom také tvrdit, že dialog je posloupnost transakcí. Obrázek 7 rozšiřuje obrázek 6 a ukazuje, které zprávy náleží jednomu dialogu.

Pomocí *CSeq* snadno rozpoznáme zprávy dialogu. Dialogy usnadňují řízení komunikace, protože UA nedrží stavy dialogu. Například zpráva INVITE zahajuje dialog a zahajuje spojení, zpráva BYE patří do dialogu, protože ukončuje spojení, viz. [8], [9].

7 Registrace a autentizace

Uživatelé se musí sami registrovat na Registrar serveru, aby byli dosažitelní. Registrace se skládá ze zprávy REGISTER následovanou odpovědí 200 OK, kterou posílá registrar v případě, že je registrace úspěšná. Pokud jsou registrace ověřovány, tak uživatel může dostat negativní odpověď 401 nebo 407, vysvětlující, že tato registrace není oprávněná, tento případ je na obrázku 8.

Základním prvkem bezpečnosti je autentizace, která v SIPu vzhledem tomu, že ideovým rodičem je protokol HTTP, používá schéma HTTP Digest. Ve starší verzi standartu (RFC 2543) bylo uváděno i HTTP Basic, které již ovšem podle RFC 3261 nesmí být používáno, tj. vyžadováno ani přijímáno. V rámci komunikace se ještě rozlišuje autentizace mezi uživateli (User-to-User) a mezi proxy serverem a uživatelem (Proxy-to-User). S prvním případem se setkáme nejčastěji u registrace. Registrační server je koncovým příjemcem požadavku, a proto je použita metoda User-to-User. Pokud nejsou potřebné údaje ve zprávě vyplněny, cílový klient posílá odpověď 401 Unauthorized a hlavička WWWAuthenticate obsahuje výzvu, viz. [10].



Obr. 8: Registrace

V případě, že proxy server potřebuje před zpracováním požadavku uživatele ověřit, žádá o to v odpovědi 407 Proxy Authentication Required a v hlavičce Pro-

xy-Authenticate je obsažena výzva. Klient doplní do požadavku hlavičku Proxy-Authorization s patřičnými údaji. Celá procedura úspěšné registrace je pochopitelně zakončena konečnou odpovědí 200 OK.

Status-Line: SIP/2.0 **401 Unauthorized**
Via: SIP/2.0/UDP
195.113.150.114;rport=5060;branch=z9hG4bKc37196720000000b4536f34100 0
From: <sip:voznak@cesnet.cz>;tag=710614011481
To: <sip:voznak@cesnet.cz>;tag=c10ed4fff3e6fb17efd0bfbdcce87ce2.7c9b
Call-ID: 67C3FC83-B95C-4947-BAEB-12B222752CDB@195.113.150.114
CSeq: 1 REGISTER
WWW-Authenticate: Digest realm=„cesnet.cz“,
nonce=„4536f4663721fbab2737d9f7131d2b4b9082b163“

Zdroj pak zopakuje požadavek s ověřovacími údaji odpovídajícími výzvě v hlavičce Authorization.

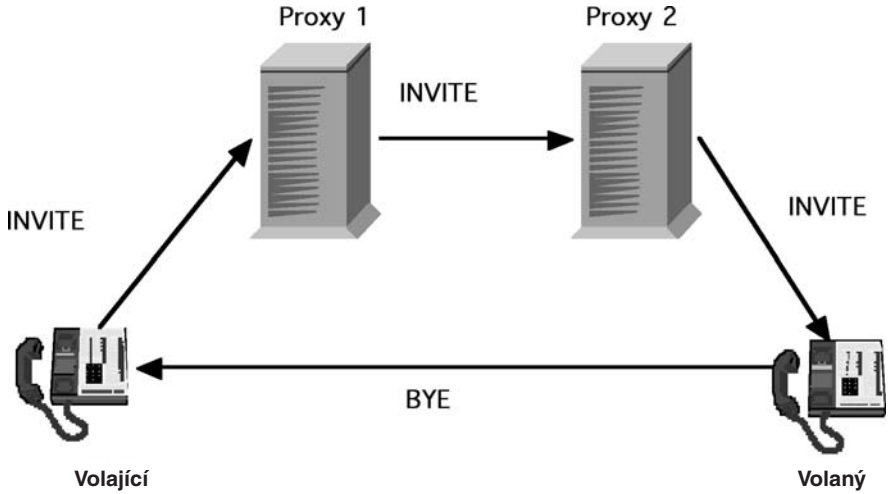
Via: SIP/2.0/UDP
195.113.150.114;rport;branch=z9hG4bKc37196720000000b4536f341000
From: <sip:voznak@cesnet.cz>;tag=710620356
To: <sip:voznak@cesnet.cz>
Contact Binding: <sip:voznak@195.113.150.114>
Call-ID: 67C3FC83-B95C-4947-BAEB-12B222752CDB@195.113.150.114
CSeq: 2 REGISTER
Authorization: Digest username="voznak",realm="cesnet.cz", nonce="
4536f4663721fbab2737d9f7131d2b4b9082b163", uri="sip:cesnet.cz", respon-
se="6bc245acb462a5d319ccefe9cb5bec3a"

8 Směrování spojení

Dialogy jsou užívány ke směrování zpráv mezi agenty. Předpokládejme, že uživatel *sip:bob@a.com* chce volat uživatele *sip:pete@b.com*. Zná SIP adresu volaného *sip:pete@b.com* , ale tato adresa jasně nevypovídá o jeho aktuální lokalizaci, volající v prvé řadě odesílá žádost INVITE na Proxy server.

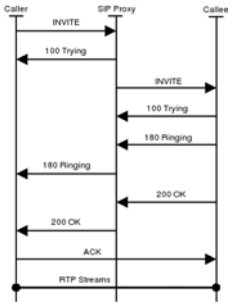
Žádost bude odesílána z jednoho Proxy na druhý, dokud nebude volaný lokalizován, což je proces směrování. UA volajícího odešle odpověď volanému a uvnitř pole *Contact* bude obsažena jeho lokalizace. Původní žádost rovněž obsahuje

pole *Contact* a tak oba UA mají informaci o své poloze. Protože oba UA už znají svou polohu, není nutné posílat další žádosti na Proxy a mohou být zasílány přímo mezi UA.



Obr. 9: Směrování

Další zprávy dialogu jsou zasílány přímo, Proxy nedostává všechny zprávy dialogu, přímo směrované zprávy jsou doručeny s podstatně menším zpožděním, protože typická Proxy obsahuje složitou směrovací logiku. Obrázek 9 obsahuje příklad zprávy uvnitř dialogu (BYE) , která obchází Proxy.



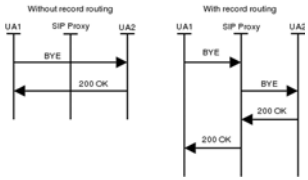
Obr. 10: INVITE

Už jsme se zmínili, že identifikátory dialogu obsahují tři části *Call-Id, From* a *To. Call-ID* je také nazýván jako identifikátor hovoru. Musí to být jedinečný řetězec znaků identifikující jedno spojení, které obsahuje jeden nebo více dialogů. Pokud se jednoho spojení účastní více UA, například v případě větvení na Proxy, tak odpovídají na žádosti a každý UA odesílá 2xx a vytváří separátní dialog s volajícím. Všechny tyto dialogy jsou ale součástí jednoho hovoru a mají stejné *Call-ID*. Do pole *From* je generováno volajícím označení (*tag*) jako jedinečný identifikátor dialogu. Do Pole *To* je označení vytvořené volaným jako jedinečný identifikátor v dialogu. Takto zvolené označení identifikátorů hovoru je nutné, protože jednoduchá inicializace spojení INVITE může znamenat několik dialogů s odpovědí a volající musí být schopen mezi nimi rozlišit.

Zahájení spojení obsahuje jednu žádost INVITE obvykle odeslanou na Proxy. Proxy server okamžitě odpovídá zprávou 100 Trying, což pro volajícího znamená, aby neopakoval žádost, že je požadavek přeposílán dál. Všechny informativní odpovědi generované volaným jsou posílané volajícímu, např. zpráva 180 Ringing, viz. obr. 10.

Zpráva 200 OK je generována ve chvíli, kdy volaný vyzvedne a je opakována, dokud odesílající UA neobdrží ACK od volajícího, v té chvíli je možné považovat spojení za sestavené. Spojení je ukončeno odesláním žádosti BYE v rámci dialogu zahájeného zprávou INVITE. Zprávy BYE jsou odesílány přímo z jednoho UA druhému UA, jestliže Proxy nepotřebuje zaznamenávat směrování. Účastník spojení, který zavěsí, odesílá zprávu BYE a z druhé strany, která se účastní spojení, přichází odpověď 200 OK, ta potvrzuje zprávu BYE a spojení je ukončeno, viz. obrázek.

Veškeré požadavky odeslané v rámci dialogu jsou standardně odeslány přímo z jednoho UA jinému. Je několik situací, ve kterých SIP Proxy potřebuje sledovat všechny zprávy, například Proxy když spolupracuje s NAT nebo pokud zajišťuje účtování, tak musí dostávat zprávu BYE. Mechanizmus, kterým Proxy může informovat UA, že si přeje dostávat všechny další zprávy spojení, se nazývá *směrování se záznamem.* Proxy server vloží do hlavičky SIP zprávy pole *Record-Route*, kde udá svoji adresu a všechny zprávy v rámci tohoto dialogu jsou posílány na SIP Proxy. Pokud příjemce žádosti obdrží řadu polí *Record-Route* , tak je musí do odpovědi rovněž zařadit, protože odesílatel je rovněž potřebuje znát.

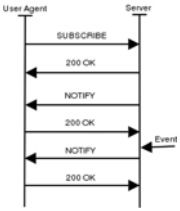


Obr. 11: BYE zpráva (směrování se záznamem a bez záznamu)

Na obrázku vlevo je zobrazení zpráv, které jsou směrovány přímo a vpravo je dialog, ve kterém se pracuje s polem *Record-Route*, což zásadně mění situaci ve směrování. U staršího RFC2543 se *směrování se záznamem* provádělo pomocí změny *Request-URI*, které bylo přepsáno. To znamená, že *Request-URI* vždy obsahovalo URI dalšího skoku (což může být jiný Proxy server nebo cílový UA). Jako poslední záznam pole *Route (*pro směrování) muselo být vloženo vždy původní *Request-URI.* Tomuto způsobu se říká přesně vymezené směrování (*strict routing)*. Volné směrování (*Loose routing)* je specifikován v RFC3261 a pracuje trošku odlišným způsobem. *Request-URI* není přepisováno a vždy obsahuje URI cílového UA. Jestliže pole *Route* v hlavičce obsahuje nějaký záznam, tak zpráva je poslána na URI nejvýše uvedeného záznamu (první v pořadí). Přechod mezi oběma typy směrování vyžaduje zpětnou kompatibilitu, především pro starší UA a to bývá častým zdrojem problémů.

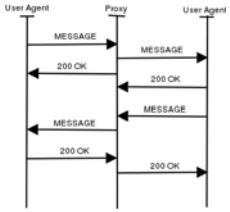
9 Prezence a Instant Messaging

Specifikace SIPu byla rozšířena o podporu mechanizmů umožňující se přihlásit k odběru určitých informací a následně zasílat události jako jsou například výměny statistik se SIP Proxy, informace o osobě, změny spojení, atd... Mechanizmus je užíván hlavně k zprostředkování informací o osobách (např. ochotných komunikovat).



Obr. 11: Popis událostí a oznamování

Obrázek 11 ukazuje základní tok zpráv. UA, který chce dostávat oznámení, tak posílá zprávu SUBSCRIBE na SIP server. Zpráva SUBSCRIBE zahajuje dialog a server okamžitě odpovídá 200 OK. Od této chvíle je dialog sestaven a server zasílá žádost NOTIFY pokaždé, jestliže se změní událost, kterou chce uživatel sledovat. Zprávy NOTIFY jsou zasílány v rámci dialogu zahájeného zprávou SUBSCRIBE. Na obrázku je možné zaznamenat, že první zpráva NOTIFY je odeslána bez ohledu na událost spouštějící oznámení. Přihlášení k oznamování stejně jako registrace má omezenou platnost, časově limitovanou a musí být periodicky obnovována. Zasílání naléhavých zpráv používá žádost MESSAGE. Zpráva MESSAGE nesestavuje dialog a text naléhavé zprávy je přenášen uvnitř SIP žádosti.



Obr. 12: Naléhavé zprávy

10 Využití DNS pro adresaci

Jak již bylo zmíněno, SIP entity jsou identifikovány použitím *SIP URI* (Uniform Resource Identifier), SIP URI má formát sip:user@host. SIP je vázán na doménu, to je důležitá vlastnost protokolu, pokud SIP Proxy obdrží INVITE s uživatelem jehož doménu nezná, může se dotázat DNS a zpravidla dostane zpět adresu SIP Proxy, která doménu obsluhuje. V DNS může být tento účel vytvořen SRV záznam. *SRV záznam* (service record) v DNS obsahuje specifické informace o dostupných službách, je definován v RFC 2782 z roku 2000. SRV záznam poskytuje následující informace:

- Service, symbolické jméno požadované služby,
- Protocol, obvykle TCP nebo UDP,
- Domain name, název domény pro níž je záznam platný,
- TTL: pole pro expiraci,
- Class: pole pro třídu (vždy IN),
- Priority: priorita pro cíl, nižší číslo bude dřív vybráno
- Weight: váha priority záznamu, při více spojení se stejnou prioritou je brána dříve v potaz vyšší váha

- Port: TCP nebo UDP port, na kterém je služba dostupná,
- Target: název stroje (hostname) poskytujícího službu.

SRV záznam může vypadá následovně:

_sip._tcp.example.com. 86400 IN SRV 0 5 5060 sipserver.example.com.

SRV nám vyřeší doménové jména, ale problém máme s telefonními čísly, protože SIP URI je vázáno na doménu a pokud bychom nahradili položku *username* telefonním číslem (e164), tak stále máme problém, jak se dovolat mimo svou doménu anebo přesněji, jak zjistit, která SIP Proxy obslouží volání na žádané číslo. Tento problém vyřešil ENUM. Standard ENUM přináší mapování telefonních čísel E.164 do doménových jmen. Lze ho chápat jako most mezi číselnými identifikátory telefonních sítí definovaných v ITU-T E.164 a jmennými identifikátory URI uložených v záznamech DNS a užívanými v Internetu. V roce 2000 byl ENUM poprvé popsán v RFC 2916, autorem byl švédský inženýr Patrik Fältström. Nápad navázat telefonní čísla do DNS vzbudil velký zájem, jelikož protokol IETF SIP od původní verze přímo používá identifikátory URI (Uniform Resource Identifiers) a ITU-T H.323 od druhé verze umožňuje používání URI. Diskuze mezi odbornou veřejností vedla k další verzi ENUM v RFC 3761 z dubna roku 2004 a nahradila předchozí specifikaci protokolu z roku 2000. Pro ENUM se užívá doména e164.arpa a ta je delegována se souhlasem zástupce státu v ITU-T na organizaci, která se stává správcem národní domény. V případě ČR je to subdoména 0.2.4.e164.arpa a správcem této domény je CZ NIC.

Standard ENUM (tElephone NUmbers Mapping) se vytváří ve spolupráci mezinárodní telekomunikační unie ITU a organizace Internet Engineering Task Force IETF. Cílem této spolupráce je standard na mapování telefonních čísel do systému doménových jmen DNS (Domain Name System), který by byl použitelný ke komerčním účelům. Tento standard je klíčovým prvkem pro konvergenci sítí založených na protokolu IP (Internet Protocol) a tradičních telefonních sítí s přepínáním okruhů PSTN (Public Switched Telephone Network). První část služby ENUM tvoří čísla ITU-T doporučení E.164. Za tímto účelem byla v ITU-T založena studijní skupina (SG2), která pracuje na administrativních postupech týkajících se služby ENUM. Druhou část vytváří organizace IETF, kde tato služba vznikla a spočívá v začlenění tohoto standardu do struktury systému doménových jmen DNS. V souvislosti s IP telefonní je hlavním cílem ENUM zprostředkovat

identifikaci a adresovat klienta služby přes internet na základě univerzálního identifikátoru, kterým bude telefonní číslo. V dubnu roku 2004 byl standard ENUM definován v dokumentaci IETF RFC 3761 The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM), viz. [11], [12].

Princip standardu ENUM spočívá v mapování telefonního čísla podle E.164 na řetězec znaků URI, který slouží k identifikaci služby nebo zdrojů, jako jsou dokumenty, soubory, maily a prakticky veškeré elektronické adresy v internetu prostřednictvím systému doménových jmen. Například http://www.enum.nic.at je URI pro rakouské webové stránky pro projekty ENUM. Výsledkem mapování telefonního čísla podle E.164 do systému DNS je naformátované telefonní číslo zapsané v opačném pořadí číslic navzájem oddělených tečkami, vložené do subdomény e164.arpa. Tato subdoména byla vytvořena pro účely standardu ENUM. Samotné mapování telefonního čísla na URI vypadá následovně:

- z použitého telefonního čísla v mezinárodním formátu např. „+420596991699“ odstraníme všechny znaky kromě číslic,
- mezi jednotlivé číslice se vloží tečky „4.2.0.5.9.6.9.9.1.6.9.9“,
- číslice se zapíší v opačném pořadí „9.9.6.1.9.9.6.9.5.0.2.4“,
- výsledný tvar se vloží do domény e164.arpa „ 9.9.6.1.9.9.6.9.5.0.2.4.e164.arpa“,
- do systému DNS se vyšle dotaz na NAPTR záznam pro příslušné doménové jméno,
- systém DNS nám vrátí příslušnou URI pro požadovanou službu např. „sip:420596991699@cesnet.cz“.

Pro praktickou realizaci aplikace je nutné, aby některý z prvků sítě VoIP prováděl ENUM *resolving*. Tuto funkci mohou plnit buď koncoví klienti nebo SIP server. V současné době je z praktického hlediska pro ENUM resolving nejvýhodnější použití SIP serveru. Nejznámější open source aplikace, které umožňují službu SIP serveru s podporou ENUM resolvingu jsou SER (SIP Expres Router) a Asterisk, které pracují pod systémem Linux. Na obrázku je znázorněn průběh adresace pomocí ENUM, aby DNS server prováděl překlad adres pro službu ENUM, musí podporovat NAPTR (The Naming Authority Pointer) záznamy. V současnosti nejpoužívanější program pro službu DNS serveru, který splňuje tuto podmínku je deamon BIND (Berkeley Internet Name Domain), aktuálně verze 9. Jako koncového klienta lze využít softwarový nebo hardwarový telefon pracující s protokolem SIP. DNS server musí obsahovat příslušný záznam, např.

```
$ORIGIN 9.9.6.1.9.9.6.9.5.0.2.4.e164.arpa.
\\IN NAPTR 100 10 „u“ „E2U+sip“ „!^+(.*)$!sip:1@cesnet.cz!“  .
\\IN NAPTR 200 10 „u“ „E2U+h323“  „!^+(.*)$!h323:1@gk1ext.cesnet.cz!“
.
```

Formát NAPTR je předepsán ve tvaru:

label IN NAPTR order pref #„u“ „E2U+enumservice“ regexp .

Tvar obsahuje pořadí (order), upřednostnění (pref) a aplikující se regulární výraz (regexp). Výsledkem je URI adresa.ENUM definuje E2U pro službu, která je očekávána pro:

- SIP jako E2U+sip, např. s URI sip:miroslav.voznak@sip.vsb.cz,
- H.323 jako E2U+h323, např. s URI h323:miroslav.voznak@h323.vsb.cz,
- Internet FAX jako E2U+ifax, např. s URI mailto:fax@fax.vsb.cz,
- Telephone jako E2U+tel, např. s URI tel:+596991699:svc=voice,
- Fax jako E2U+fax:tel, např. s URI tel:+596991699:svc=fax,
- Email jako E2U+email:mailto, např. s URI mailto:miroslav.voznak@vsb.cz,
- Web jako E2U+web:http, např. s URI http://www.vsb.cz.

Pro vyhledání NAPTR můžeme v Linuxu aplikovat příkazy:

```
host -t naptr 9.9.6.1.9.9.6.9.5.0.2.4.e164.arpa
dig -t naptr NUMMER.e164.arpa.
```

Obdobně pro vyhledání SRV záznamů můžeme v Linuxu aplikovat příkaz:

```
host -t SRV _sip._udp.cesnet.cz
host -t SRV _sip._tcp.cesnet.cz
```

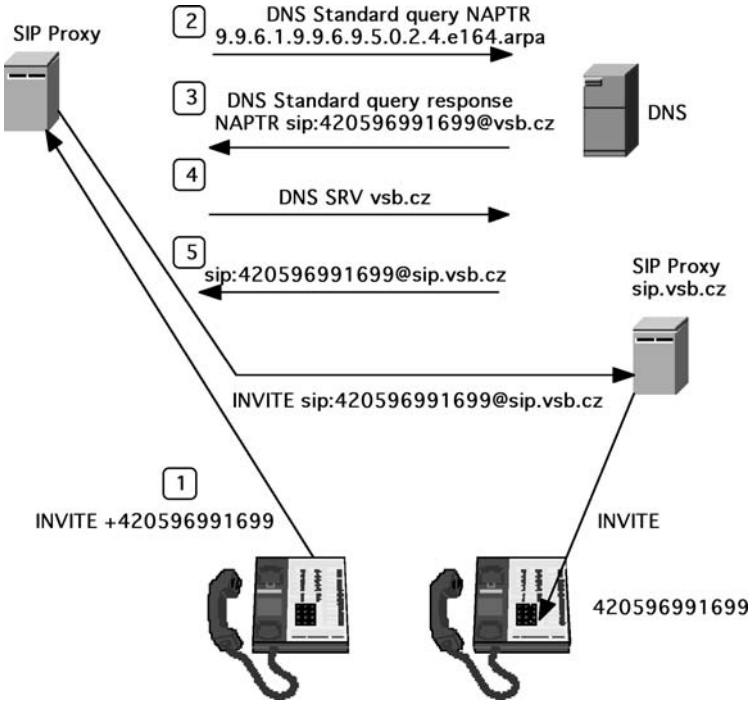
V případě Windows je použitelný příkaz *nslookup* pro vyhledání SRV, bohužel *naptr* nezná:

```
nslookup
>set q=srv
>_sip._udp.cesnet.cz
```

Kromě veřejného stromu e164.arpa je možné si vytvořit libovolný privátní strom, např. e164.cesnet.cz, ale číslo bude dosažitelné pouze uvnitř instituce ane-

bo klientům, kteří se budou takovéto DNS dotazovat, což jde ošetřit na úrovni SIP Proxy nebo koncového zařízení. Privátní stromy nezaručují obecnou dostupnost a omezují se na skupinu nebo instituci. Pro více stromů musí být pochopitelně více dotazů a dochází k štěpení, proto je žádoucí používat e164.arpa. Na obrázku je adresace klienta s použitím ENUM. Vidíme, že klient odeslal INVITE s tel.č., z DNS byla vrácena SIP URI, která pak umožnila nalézt cílový SIP PROXY (pomocí SRV).

ENUM můžeme použít jako vyhledávací mechanizmus na osobním počítači, protože na základě univerzální identifikace (telefonního čísla) získáme informace o všech možných přístupech k požadovanému subjektu. Do databáze služby ENUM budou zařazena i negeografická čísla. Společnosti tak mohou používat jako adresu svých webových na místo doménového jména, které bývá často špatně zapamatovatelné např. číslo *Zelené linky*. Koncoví klienti sítě VoIP nemusí mít vlastní telefonní číslo, ale mohou použít číslo přidělené pevné lince.



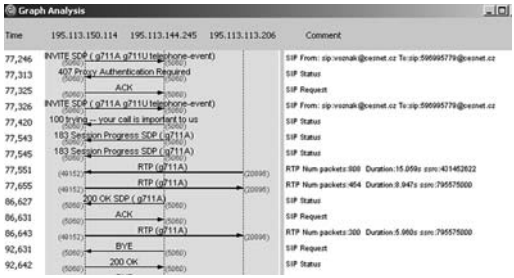
Obr. 13: Adresace klienta SIP v síti VoIP pomocí ENUM

Možnosti služby ENUM jsou rozsáhle a lze očekávat, že praxe najde další aplikace pro využití tohoto standardu. Aby bylo možné zahájit zkušební projekty ENUM je nutné zajistit možnost registrace. V červnu roku 2003 byla delegována na CZ NIC česká doména ENUM **0.2.4.e164.arpa** a následně sdružení CESNET začalo experimentovat se standardem ENUM v rámci své akademické sítě. V současnosti je CESNET největším držitelem čísel alokovaných v doménách, jedná se o cca 200 tis. čísel.

11 Praktická ukázka SIPu

Zachycení jakékoliv komunikace na IP a její rozklíčování umožňuje nepřeberné množství aplikací jak pro Windows, tak i pro Linux. Bezkonkurenčně nejlepším nástrojem v poměru ceny a výkonu je aplikace Wireshark volně šiřitelná pod licencí GNU GPL. Wireshark přímo navazuje na Ethereal, je jeho následovníkem. Mnozí možná znají označení Ethereal, v polovině roku 2006 odešel původní autor Etherealu pracovat do nejmenované komerční firmy a nečekaně vzal sebou i registrovanou značku Ethereal. Pro zachování open source vývoje bylo nutné změnit název.

Na obrázku 14. je znázorněna komunikace v grafu, ta se získá přes menu Statistics -> VoIP Calls, v okně se zobrazí všechny zachycené VoIP spojení, po označení lze vybrat položku graph a získat přehledně označený tok zpráv. Následuje rozparsování jenotlivých zpráv, ale pouze SIP bez SDP. Na obrázku 14 lze vidět, ve kterých zprávách se navíc přenáší SDP, výsledkem je dohoda na kodeku G.711 A law a UDP portech komunikujících stran, tyto položky jsou v SDP označeny jako Media format a Media port. RTP pakety jsou přenášeny už před přihlášením (200 OK), jelikož byl poslán 183 Session Progress, který obsahoval část SDP. Komunikace začala žádostí INVITE, která ovšem vyžadovala autentizaci na SIP Proxy (407 Authentication Required), po opakovaném odeslání INVITE rozšířeném o pole Proxy-Authorization, byl přijat 100 Trying. Stateful Proxy odesílá 100 Trying okamžitě po přijetí INVITE a nečeká na 100 Trying od další SIP Proxy nebo UA, později přijatý 100 Trying už dále nepřeposílá. Jedná se o případ tohoto spojení, použitá SIP Proxy je SIP Express Router, což je i čitelné z rozparsovaných zpráv. Jak už bylo zmíněno, tak 183 Session Progress nese i část SDP a následně mohou být přenášeny RTP pakety již před přihlášením. Po přihlášení přichází konečná odpověď 200 OK potvrzená žádostí ACK. Ukončení spojení proběhlo ze strany volaného žádostí BYE, která byla potvrzena konečnou odpovědí 200 OK. Následuje rozparsovaný sled zpráv SIP vyměněných během spojení.



Obr. 14: Adresace klienta SIP v síti VoIP pomocí ENUM

Request-Line: **INVITE** sip:596995779@cesnet.cz SIP/2.0
Method: INVITE
Resent Packet: False
Message Header
Via: SIP/2.0/UDP
195.113.150.114;rport;branch=z9hG4bKc3719672000000194536f381000006f20
0000005
From: "unknown"<sip:voznak@cesnet.cz>;tag=717026516927
SIP Display info: "unknown"
SIP from address: sip:voznak@cesnet.cz
SIP tag: 717026516927
To: <sip:596995779@cesnet.cz>
SIP to address: sip:596995779@cesnet.cz
Contact: <sip:voznak@195.113.150.114>
Contact Binding: <sip:voznak@195.113.150.114>
URI: <sip:voznak@195.113.150.114>
SIP contact address: sip:voznak@195.113.150.114
Call-ID: 34717910-B1CD-4563-9301-481F2F702E24@195.113.150.114
CSeq: 1 INVITE
Max-Forwards: 70
User-Agent: SJphone/1.61.321a (SJ Labs)
Content-Length: 246
Content-Type: application/sdp
Message body

Status-Line: SIP/2.0 **407 Proxy Authentication Required**
Status-Code: 407

```
        Resent Packet: False
      Message Header
        Via: SIP/2.0/UDP
195.113.150.114;rport=5060;branch=z9hG4bKc3719672000000194536f381000
006f200000005
        From: "unknown"<sip:voznak@cesnet.cz>;tag=717026516927
          SIP Display info: "unknown"
          SIP from address: sip:voznak@cesnet.cz
          SIP tag: 717026516927
        To:
<sip:596995779@cesnet.cz>;tag=c10ed4fff3e6fb17efd0bfbdcce87ce2.be11
          SIP to address: sip:596995779@cesnet.cz
          SIP tag: c10ed4fff3e6fb17efd0bfbdcce87ce2.be11
        Call-ID: 34717910-B1CD-4563-9301-481F2F702E24@195.113.150.114
        CSeq: 1 INVITE
        Proxy-Authenticate: Digest realm="cesnet.cz",
nonce="4536f4a73fe0b7da3d22aeef5150fc5259b2ab44"
        Server: Sip EXpress router (0.9.5-pre1 (i386/linux))
        Content-Length: 0
        Warning: 392 195.113.144.245:5060 "Noisy feedback tells: pid=30674
req_src_ip=195.113.150.114 req_src_port=5060 in_uri=sip:596995779@cesnet.cz
out_uri=sip:596995779@cesnet.cz via_cnt==1"

      Request-Line: ACK sip:596995779@cesnet.cz SIP/2.0
        Method: ACK
        Resent Packet: False
      Message Header
        Via: SIP/2.0/UDP
195.113.150.114;rport;branch=z9hG4bKc371967200000194536f381000006f20
0000005
        From: "unknown"<sip:voznak@cesnet.cz>;tag=717026516927
          SIP Display info: "unknown"
          SIP from address: sip:voznak@cesnet.cz
          SIP tag: 717026516927
        To:
<sip:596995779@cesnet.cz>;tag=c10ed4fff3e6fb17efd0bfbdcce87ce2.be11
          SIP to address: sip:596995779@cesnet.cz
          SIP tag: c10ed4fff3e6fb17efd0bfbdcce87ce2.be11
```

Call-ID: 34717910-B1CD-4563-9301-481F2F702E24@195.113.150.114
CSeq: 1 ACK
Max-Forwards: 70
User-Agent: SJphone/1.61.321a (SJ Labs)
Content-Length: 0

Request-Line: **INVITE** sip:596995779@cesnet.cz SIP/2.0
Method: INVITE
Resent Packet: False
Message Header
Via: SIP/2.0/UDP
195.113.150.114;rport;branch=z9hG4bKc3719672000000194536f38100003812
00000006
From: "unknown"<sip:voznak@cesnet.cz>;tag=717026516927
SIP Display info: "unknown"
SIP from address: sip:voznak@cesnet.cz
SIP tag: 717026516927
To: <sip:596995779@cesnet.cz>
SIP to address: sip:596995779@cesnet.cz
Contact: <sip:voznak@195.113.150.114>
Contact Binding: <sip:voznak@195.113.150.114>
URI: <sip:voznak@195.113.150.114>
SIP contact address: sip:voznak@195.113.150.114
Call-ID: 34717910-B1CD-4563-9301-481F2F702E24@195.113.150.114
CSeq: 2 INVITE
Max-Forwards: 70
User-Agent: SJphone/1.61.321a (SJ Labs)
Content-Length: 246
Content-Type: application/sdp
Proxy-Authorization: Digest username="voznak",realm="cesnet.cz",non-
ce="4536f4a73fe0b7da3d22aeef5150fc5259b2ab44",uri="sip:596995779@ces-
net.cz",response="88ae5edfd89f2f4ce5e9d6d267a99724"
Message body

Status-Line: SIP/2.0 **100 trying -- your call is important to us**
Status-Code: 100
Resent Packet: False
Message Header

Via: SIP/2.0/UDP
195.113.150.114;rport=5060;branch=z9hG4bKc3719672000000194536f381000
0381200000006
From: "unknown"<sip:voznak@cesnet.cz>;tag=717026516927
SIP Display info: "unknown"
SIP from address: sip:voznak@cesnet.cz
SIP tag: 717026516927
To: <sip:596995779@cesnet.cz>
SIP to address: sip:596995779@cesnet.cz
Call-ID: 34717910-B1CD-4563-9301-481F2F702E24@195.113.150.114
CSeq: 2 INVITE
Server: Sip EXpress router (0.9.5-pre1 (i386/linux))
Content-Length: 0
Warning: 392 195.113.144.245:5060 "Noisy feedback tells: pid=30676 req_src_ip=195.113.150.114 req_src_port=5060 in_uri=sip:596995779@cesnet.cz out_uri=sip:596995779@195.113.144.77:5060 via_cnt==1"

Status-Line: SIP/2.0 **183 Session Progress**
Status-Code: 183
Resent Packet: False
Message Header
Via: SIP/2.0/UDP
195.113.150.114;rport=5060;branch=z9hG4bKc3719672000000194536f381000
0381200000006
From: "unknown"<sip:voznak@cesnet.cz>;tag=717026516927
SIP Display info: "unknown"
SIP from address: sip:voznak@cesnet.cz
SIP tag: 717026516927
To: <sip:596995779@cesnet.cz>
SIP to address: sip:596995779@cesnet.cz
Date: Thu, 19 Oct 2006 03:39:38 GMT
Call-ID: 34717910-B1CD-4563-9301-481F2F702E24@195.113.150.114
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 2 INVITE
Content-Type: application/sdp
Session: Media
Content-Length: 140
Message body

Status-Line: SIP/2.0 **183 Session Progress**
Status-Code: 183
Resent Packet: False
Message Header
Via: SIP/2.0/UDP
195.113.150.114;rport=5060;branch=z9hG4bKc3719672000000194536f381000
0381200000006
From: "unknown"<sip:voznak@cesnet.cz>;tag=717026516927
SIP Display info: "unknown"
SIP from address: sip:voznak@cesnet.cz
SIP tag: 717026516927
To: <sip:596995779@cesnet.cz>
SIP to address: sip:596995779@cesnet.cz
Date: Thu, 19 Oct 2006 03:39:38 GMT
Call-ID: 34717910-B1CD-4563-9301-481F2F702E24@195.113.150.114
Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
CSeq: 2 INVITE
Content-Type: application/sdp
Session: Media
Content-Length: 140
Message body

Status-Line: SIP/2.0 **200 OK**
Status-Code: 200
Resent Packet: False
Message Header
Via: SIP/2.0/UDP
195.113.150.114;rport=5060;branch=z9hG4bKc3719672000000194536f381000
0381200000006
From: "unknown"<sip:voznak@cesnet.cz>;tag=717026516927
SIP Display info: "unknown"
SIP from address: sip:voznak@cesnet.cz
SIP tag: 717026516927
To: <sip:596995779@cesnet.cz>;tag=1EB4EC64-4F0
SIP to address: sip:596995779@cesnet.cz
SIP tag: 1EB4EC64-4F0
Date: Thu, 19 Oct 2006 03:39:38 GMT
Call-ID: 34717910-B1CD-4563-9301-481F2F702E24@195.113.150.114

```
        Server: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
        Contact: <sip:596995779@195.113.113.206:5060;user=phone>
          Contact Binding: <sip:596995779@195.113.113.206:5060;user=phone>
            URI: <sip:596995779@195.113.113.206:5060;user=phone>
              SIP contact address: sip:596995779@195.113.113.206:5060
        Record-Route: <sip:195.113.144.245;ftag=717026516927;lr=on>
        CSeq: 2 INVITE
        Content-Type: application/sdp
        Content-Length: 140
      Message body

      Request-Line: ACK sip:596995779@195.113.113.206:5060;user=phone SIP/2.0
        Method: ACK
        Resent Packet: False
      Message Header
        Via: SIP/2.0/UDP
195.113.150.114;rport;branch=z9hG4bKc3719672000000194536f38b000078da0
000000b
        From: "unknown"<sip:voznak@cesnet.cz>;tag=717026516927
          SIP Display info: "unknown"
          SIP from address: sip:voznak@cesnet.cz
          SIP tag: 717026516927
        To: <sip:596995779@cesnet.cz>;tag=1EB4EC64-4F0
          SIP to address: sip:596995779@cesnet.cz
          SIP tag: 1EB4EC64-4F0
        Call-ID: 34717910-B1CD-4563-9301-481F2F702E24@195.113.150.114
        CSeq: 2 ACK
        Max-Forwards: 70
        User-Agent: SJphone/1.61.321a (SJ Labs)
        Content-Length: 0
        Route: <sip:195.113.144.245;ftag=717026516927;lr=on>

      Session Initiation Protocol
        Request-Line: BYE sip:voznak@195.113.150.114:5060 SIP/2.0
          Method: BYE
          Resent Packet: False
        Message Header
          Record-Route: <sip:195.113.144.245;ftag=1EB4EC64-4F0;lr=on>
```

```
        Via: SIP/2.0/UDP 195.113.144.245;branch=z9hG4bK6812.3fca3643.0
        Via: SIP/2.0/UDP  195.113.113.206:5060
        From: <sip:596995779@cesnet.cz>;tag=1EB4EC64-4F0
          SIP from address: sip:596995779@cesnet.cz
          SIP tag: 1EB4EC64-4F0
        To: "unknown"<sip:voznak@cesnet.cz>;tag=717026516927
          SIP Display info: "unknown"
          SIP to address: sip:voznak@cesnet.cz
          SIP tag: 717026516927
        Date: Thu, 19 Oct 2006 03:39:48 GMT
        Call-ID: 34717910-B1CD-4563-9301-481F2F702E24@195.113.150.114
        User-Agent: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
        Max-Forwards: 5
        Timestamp: 1161229194
        CSeq: 101 BYE
        Content-Length: 0
        P-hint: rr-enforced

    Status-Line: SIP/2.0 200 OK
      Status-Code: 200
      Resent Packet: False
    Message Header
      Via: SIP/2.0/UDP
195.113.144.245;branch=z9hG4bK6812.3fca3643.0,SIP/2.0/UDP
195.113.113.206:5060
      From: <sip:596995779@cesnet.cz>;tag=1EB4EC64-4F0
        SIP from address: sip:596995779@cesnet.cz
        SIP tag: 1EB4EC64-4F0
      To: "unknown"<sip:voznak@cesnet.cz>;tag=717026516927
        SIP Display info: "unknown"
        SIP to address: sip:voznak@cesnet.cz
        SIP tag: 717026516927
      Call-ID: 34717910-B1CD-4563-9301-481F2F702E24@195.113.150.114
      CSeq: 101 BYE
      Content-Length: 0
      Server: SJphone/1.61.321a (SJ Labs)
```

Request-Line: **BYE** sip:voznak@195.113.150.114:5060 SIP/2.0
Method: BYE
Resent Packet: True
Suspected resend of frame: 1686
Message Header
Record-Route: <sip:195.113.144.245;ftag=1EB4EC64-4F0;lr=on>
Via: SIP/2.0/UDP 195.113.144.245;branch=z9hG4bK6812.3fca3643.0
Via: SIP/2.0/UDP 195.113.113.206:5060
From: <sip:596995779@cesnet.cz>;tag=1EB4EC64-4F0
SIP from address: sip:596995779@cesnet.cz
SIP tag: 1EB4EC64-4F0
To: "unknown"<sip:voznak@cesnet.cz>;tag=717026516927
SIP Display info: "unknown"
SIP to address: sip:voznak@cesnet.cz
SIP tag: 717026516927
Date: Thu, 19 Oct 2006 03:39:48 GMT
Call-ID: 34717910-B1CD-4563-9301-481F2F702E24@195.113.150.114
User-Agent: Cisco VoIP Gateway/ IOS 12.x/ SIP enabled
Max-Forwards: 5
Timestamp: 1161229194
CSeq: 101 BYE
Content-Length: 0
P-hint: rr-enforced

Status-Line: SIP/2.0 **200 OK**
Status-Code: 200
Resent Packet: True
Suspected resend of frame: 1688
Message Header
Via: SIP/2.0/UDP
195.113.144.245;branch=z9hG4bK6812.3fca3643.0,SIP/2.0/UDP
195.113.113.206:5060
From: <sip:596995779@cesnet.cz>;tag=1EB4EC64-4F0
SIP from address: sip:596995779@cesnet.cz
SIP tag: 1EB4EC64-4F0
To: "unknown"<sip:voznak@cesnet.cz>;tag=717026516927
SIP Display info: "unknown"
SIP to address: sip:voznak@cesnet.cz

SIP tag: 717026516927
Call-ID: 34717910-B1CD-4563-9301-481F2F702E24@195.113.150.114
CSeq: 101 BYE
Content-Length: 0
Server: SJphone/1.61.321a (SJ Labs)

12 Otevřená řešení pro SIP

V případě SIPu máme k dispozici několik zdařilých otevřených projektů, mezi které patří sipX na http://www.sipfoundry.org/index.html a dále SER a Asterisk. SER (SIP Express Router) je open source projekt napsaný v jazyce C. Zastává funkci registračního, redirect a proxy serveru pro komunikaci protokolem SIP. Původně byl vyvíjen jako páteřní SIP směrovač, tudíž s důrazem na výkon. Zvládá obsloužit tisíce hovorů za vteřinu na dvouprocesorovém PC a stovky hovorů na IPAQu.

Server podporuje bezestavové i transakčně stavové zpracování SIP komunikace. Toto zpracování je řízeno silným skriptovacím jazykem. Síla jazyka v sobě však skrývá i případnou složitost konfigurace. Server je modulární, tedy dobře rozšířitelný a s rostoucí uživatelskou základnou stále přibývají nové moduly. Server je možné vzdáleně řídit přes FIFO nebo UNIX socket a pro vlastní SIP komunikaci podporuje IP protokol verze 4 i 6 a transportní protokoly UDP,TCP a s příslušnými moduly i TLS. Server dále podporuje více uživatelských domén, aliasy a dotazy do ENUM domén. Autentizace a autorizace může být prováděna přes databázi (Mysql , Postgress), Radius nebo Diameter. Registrační vazby mohou být ukládány do Sql databáze. Server však nutně ke svému běhu databázovou podporu nepotřebuje. Registrační vazby pak zůstávají uloženy pouze v paměti, což je rychlé, ale při restartu serveru tyto záznamy zmizí. SER je schopen pomáhat při NAT průchodu (nathelper, rtpproxy, mediaproxy) a jeho funkcionalitu je možné rozšiřovat i pomocí externích skriptů například v Perlu. SER je k dipozici pro Linux, BSD (NetBSD, FreeBSD), Solaris. Nedávno však došlo k rozštěpení projektu, původní projekt pokračuje na http://sip-router.org/ a nová větev na http://openser.org/ . Kompatibilita mezi větvemi není nezaručena především u rozšíření jako SerWeb. SER je používán v CESNETu jako přístupový směrovač i server pro obsluhu uživatelů.

Dalším otevřeným řešením je Asterisk. Oficiálně je Asterisk open source hybrid TDM a packet voice PBX, jedná se o IVR (Interactive Voice Response) platformu s funkčností Automatic Call Distribution (ACD). Jde tedy o kompletní open source softwarovou PBX, běžící na platformách Linux a Unix, poskytující veške-

ré vlastnosti PBX. Systém je navržen tak, aby vytvořil rozhraní telefonnímu hardwaru, softwaru a libovolné telefonní aplikaci. Asterisk může být mimo jiné použit v těchto aplikacích:

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

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

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

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

Jelikož autor Asterisku (Mark Spencer) neměl v přílišné oblibě protokol H.323, rozhodl se navrhnout a realizovat svůj vlastní protokol. Výsledkem je protokol IAX (Inter Asterisk eXchange) jenž se stará o signalizaci a transport packet voice mezi dvěmi připojenými uzly. Ačkoliv jméno naznačuje přítomnost Asterisku na obou koncích komunikace, IAX může ve skutečnosti spojit každé dva koncové body podporující tento protokol. Následně byla přidána podpora součinnosti s ostatními VoIP systémy a podpora pro další protokoly, jedná se o protokoly SIP, H.323 a MGCP.

Prostřednictvím kanálů vstupují do systému různé formáty komunikace. Kanály jsou logická spojení s různými signalizačními a přenosovými cestami, které může Asterisk využívat k vytváření a spojování jednotlivých hovorů. Kanál by mohl představovat spojení s obyčejným telefonním přístrojem (telefonní linkou) nebo Internetové telefonní hovory („logické volání“). Asterisk nedělá žádný rozdíl mezi typem kanálu „FXO (Foreign eXchange Office)“ a „FXS (Foreign eXchange Station)“, nerozlišuje tedy mezi telefonními linkami a telefony. Každé volání je umístěno na odlišném kanále. Asterisk se všemi kanály zachází jako s přípojnými body, jejichž vzájemná interakce se definuje v Dialplan (extensions.conf). Je důležité si uvědomit, že i kdyby se kanály lišily v rámci použité technologie a konektivity, Asterisk umožňuje zacházet se všemi jakoby byly téměř stejné.

V současnosti existuje několik nezávislých implementací kanálu H.323 do systému Asterisk (h323, oh323, ooh323c, woomera). V případě IAX se konfigurace kanálů provádí modifikací souboru iax.conf. Kanál chan_local je pseudo-kanál. Používá se pro vytvoření smyčky, která volá zpět do Dialplan v různých context. Užitečné je především rekurzivní směrování, které je schopno vracet se do Dialplan po ukončení volání. SIP kanálový modul umožní Asterisku VoIP komunikaci se SIP telefony a ústřednami.

Konfigurace SIP kanálů/klientů se provádí modifikací souboru sip.conf. Zap kanálový modul poskytne mezivrstvu (interface layer) mezi Asteriskem na jedné straně a Zaptel a/nebo ZapHFC ovladači rozhraní na straně druhé. Konfigurace ZAP kanálů se provádí modifikací souboru zapata.conf.

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

Literatura

[1] Camp, K: *IP Telephony Demistified.* McGraw-Hill, New York 2003, ISBN 0-07-140670-0.

[2] Vodrážka, J. – Pravda, I.: *Principy telekomunikačních systémů*, nakladatelství ČVUT,2006. ISBN 80-01-03366-X.

[3] Hardy,W.: *VoIP service quality.* McGraw-Hill, New York, 2003, ISBN 0-07-141076-7.

[4] ITU-T P.861: *Objective quality measurement of telephoneband (300-3400 Hz) speech codecs*. Geneva, May 2000

[5] H.323v5: *Recommendation H.323*,ITU-T, July 2003, http://www.itu.int/rec/T-REC-H.323/en

[6] Collins, D.: *Carrier Grade Voice Over IP*. McGraw-Hill, 2002, ISBN 0071406344.

[7] Peters, J. – Davidson, J.: *Voice over IP Fundamentals*. Cisco Press, Indianopolis,2000, ISBN 1-57870-168-6.

[8] RFC 3261: *SIPSession Initiation Protocol*, IETF, http://www.unix.org.ua/rfc/rfc3261.html, June 2002,

[9] Baroňák, I. – Halás,M.: *SIP Protocol – the Future of IP Telephony*, TSP 2004, Brno 2003, ISBN 80-214-2684-5.

[10] Vozňák, M. – Růžička, J.: Bezpečnost v sítích s VoIP, 5/2006, TaP, Wirelesscom Praha, Květen 2006, ISSN 1213-7162

[11] Vozňák, M. – Machálek, P.: ENUM, Connect!, květen 2005, Computer Press Praha, ISSN 1211-3085

[12] Vozňák, M. : Co je nového v protokolu SIP, Connect! 1/2005, Computer Press Praha, leden 2005, ISSN 1211-3085