ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta elektrotechnická
Katedra telekomunikační techniky

LABORATORNÍ SÍŤ IMS

Leden 2016

Autor: **Bc. Jan Ludvík**

Vedoucí práce: **Ing. Pavel Troller, CSc.**

Čestné prohlášení

Prohlašuji, že jsem zadanou diplomovou práci zpracoval sám s přispěním vedoucího práce a používal jsem pouze literaturu v práci uvedenou. Dále prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé diplomové práce nebo její části se souhlasem katedry.

V Praze 11.1.2016

……

Poděkování

Děkuji vedoucímu práce Ing. Pavlu Trollerovi, CSc. za veškerou pomoc a konzultace, které vždy vyjasnily všechny problémy a posunuly mě v tématu dál.

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra telekomunikační techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Ludvík Jan**

Studijní program: Komunikace, multimédia a elektronika
Obor: Sítě elektronických komunikací

Název tématu: **Laboratorní síť IMS**

Pokyny pro vypracování:

Navrhněte, z dostupných komponent s otevřeným kódem sestavte a nakonfigurujte laboratorní experimentální síť standardu IMS. Tato síť by kromě jádra (P-CSCF, S-CSCF, I-CSCF a HSS), dostupného např. v projektu OpenIMSCore, měla obsahovat též doplňující komponenty, jako je Aplikační server (AS), Media server (MRF), Breakout Gateway (BGW) a případné další komponenty, umožňující testovat pokročilé scénáře spolupráce IMS subsystému s pre-IMS platformami a systémy, s mobilními sítěmi (včetně možnosti testování VoLTE) a s různými typy uživatelských zařízení (UEs).
Pohledejte dostupné komponenty dostupné pod licencemi s otevřeným kódem a případně navrhněte možnost upravit pro tyto potřeby jiné existující projekty (např. Asterisk, Kamailio atd.).

Seznam odborné literatury:

[1] Camarillo, Gonzalo; García-Martín, Miguel A. (2007). The 3G IP multimedia subsystem (IMS) : Merging the Internet and the Cellular Worlds (2 ed.). Chichester [u.a.]: Wiley. ISBN 0-470-01818-6
[2] Syed A. Ahson, Mohammed Ilyas, ed. (2009). IP multimedia subsystem (IMS) handbook. Boca Raton: CRC Press. ISBN 1-4200-6459-2.

Vedoucí: Ing. Pavel Troller, CSc.

Platnost zadání: do konce letního semestru 2015/2016

L.S.

prof. Ing. Boris Šimák, CSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 19. 11. 2014

Anotace:

S postupným rozvojem sítí LTE se do praxe dostává kromě vyšších přenosových rychlostí také možnost využít pakety pro přenos hlasu. Síť IMS je standardem, od kterého se odvíjí práce v oblasti VoLTE v komerční sféře. Nejedná se však jen o síť pro přenos hlasu, ale všeho multimediálního obsahu na všech přístupových technologiích, které dnes známe. V budoucnu lze očekávat konvergování různých komunikačních platforem právě do sítě, která bude z velké části vyhovovat právě standardům IMS od 3GPP.

Klíčová slova:

IMS, SIP, VoLTE, telefonní síť

Abstract:

LTE networks are being deployed all over the world and that enables ISPs to not only offer higher data speeds, but also exploit low latency of data to provide voice service over provider's network. IMS network is the foundation on which all VoLTE systems are bulit. IMS network strives to offer not just voice or audio transfer, but also all types of data a people might want to exchange with each other while disregarding any dependencies on access technologies. It is very likely that we will see different communication platforms converge under one telephone network in the future based on this very standard which is IMS from 3GPP.

Key words:

IMS, SIP, VoLTE, telephone network

Obsah

1 Úvod

Hlasové služby zažívají v poslední době poměrně velký rozvoj. Po letech, kdy veškerá komunikace probíhala na základě spojování okruhů v síti operátora, se postupně začínají objevovat paketově orientované sítě. Zmíněný rozvoj probíhá u nás tradičně se zpožděním za zbytkem vyspělého světa, ale v případě těchto technologií ne až tak markantním. První VoLTE (*Voice over LTE* – LTE (*Long term evolution*)), jenž je výsledkem vývoje IMS (*IP Multimedia Subsystem*), byl komerčně spuštěn v Singapuru v květnu 2014 [1]. Český T-mobile následoval zbytek světa jako první v ČR o rok později, 4. 5. 2015 [2], další čeští operátoři zatím VoLTE pouze testují.

Velkou výhodou CS (circuit switched) sítí byla dlouho bezkonkurenční kvalita hovoru. V sítích 3G a starších data nemohla hlasu konkurovat především v odezvě, která byla pro jakýkoliv transport hlasu naprosto nevyhovující. To se však změnilo s uvedením sítě označované 4G (toto označení bylo poměrně sporné, původně podle 3GPP (*3rd Generation Partnership Project*) se jako 4G označovalo až LTE Advanced a LTE mělo být 3,5G – marketingu operátorů však pochopitelně s novou sítí více vyhovovalo označení LTE jako 4G). LTE nabízí několik výhod oproti starším sítím – pro uživatele mobilního internetu je největším posunem hlavně mnohem větší datová propustnost. Pro interaktivní služby typu hlasových, případně videohovorů, je však nejzásadnější už zmíněná odezva, která se pohybuje u LTE kolem jednotek milisekund [citace], což je už pro kvalitní hlasový hovor dostatečné. Přechodu na přenos hlasu v paketech už tedy nic z technologického hlediska nebránilo.

S prvním nasazením LTE se však nepřešlo okamžitě také na VoLTE. V prvních fázích nasazení LTE se běžně pro spojení hovorů využívá takzvaný CS fallback, kdy dojde k přepnutí telefonu do starších sítí. Velmi nepříjemným vedlejším efektem je odpojení dat v okamžiku přepnutí na celou dobu hovoru a tedy ukončení všech datových přenosů, kterých, samozřejmě podle uživatelových zvyků a aplikací, nemusí být rozhodně málo.

Z vlastních zkušeností a také ohlasů okolí mohu potvrdit, že toto přepnutí na některých telefonech u jednoho operátora ne vždy probíhalo dobře a po ukončení hovoru ne vždy došlo k obnovení LTE (k obnovení přenosu dat). Toto je samozřejmě chyba, která byla později odstraněna.

IMS síť jako taková není nasazována v přesné podobě, jakou určuje 3GPP v jednotlivých releasech i v reálném životě. Samozřejmě některé standardy nelze změnit kvůli operátorské interoperabilitě, velké části VoLTE komerčních řešení jednotlivých firem jsou však dosti jiné. V mé práci jsem využil pouze open-source zdroje, které jsou vyvíjeny komunitou. Bohužel však vývoj v této oblasti není velmi aktivní, dalo by se říci, že na rozvoji pracuje pouze několik jednotlivců a testování a použití se odehrává hlavně ve výzkumech vysokých škol, případně je open-source IMS využito jako testovací platforma u operátorů a firem. Konkrétnější popis jednotlivých použitých systémů a projektů bude následovat v dalších kapitolách diplomové práce.

2 Síť IMS

IMS je projekt organizace 3GPP. První návrhy byly obsaženy v Release 1999, který datuje do roku 1999. Během dalších let následovalo několik dalších Release, vždy přidávající další funkcionality nebo měnící starší koncepty, další rozvoj sítě je obsažen jak v Release 12, tak i Release 13, v plánovaném Release 14 zatím mnoho informací o IMS není k dispozici [3].

Hlavní myšlenkou celé sítě je rozvoj od CS sítí k PS, tedy nahrazení starších signalizačních protokolů především signalizačním protokolem SIP (*Session Initiation Protocol*), který je široce využit v IMS pro komunikaci mezi zařízeními i dalšími jednotlivými bloky. IMS následuje architekturu rozdělenou do několika vrstev, každá zajišťuje jiné funkce a v každé vrstvě je obsaženo několik bloků, které zajišťují určité funkce systému.

První vrstva je přístupová vrstva (nazývána Transport layer) [4] a zajišťuje připojení jednotlivých zařízení do sítě. Druhá vrstva se stará o veškerý routing signalizace a obsahuje všechny důležité prvky pro spojení hovoru a různé brány pro spojení do jiných typů sítí. Nejvyšší vrstva pak má na starosti především správu údajů o všech uživatelích v síti (nahrazuje v podstatě všechny dosavadní databáze a registry všech možných druhů sítí) a také aplikační servery, s jejichž pomocí síť může poskytovat širokou škálu služeb – jako příklad lze uvést konferenční hovory nebo komunikátory typu IM (*Instant Messaging*) a mnoho dalších.

Nejdůležitější vlastností IMS, kromě vývoje od CS sítí k PS, je naprostá nezávislost na přístupové metodě. Nezáleží, jestli uživatel je na WIFI (Wireless Fidelity), LTE, připojen optickým kabelem, přes xDSL (*Digital Subscriber Line*) nebo WiMAX (Worldwide Interoperability for Microwave Access). Každý uživatel, který má v síti svoji identitu, může provádět stejné úkony. Uživatel síť využívá pomocí jeho UE (*User Equipment*). Další významné vlastnosti budou uvedeny v samostatné kapitole.

Komunikace mezi bloky IMS probíhá přes rozhraní (interface). Každý funkční blok může komunikovat s několika jinými bloky a každá možnost komunikace má vlastní rozhraní s vlastním označením. Významně to ulehčuje popis funkce sítě.

2.1 Vrstvy

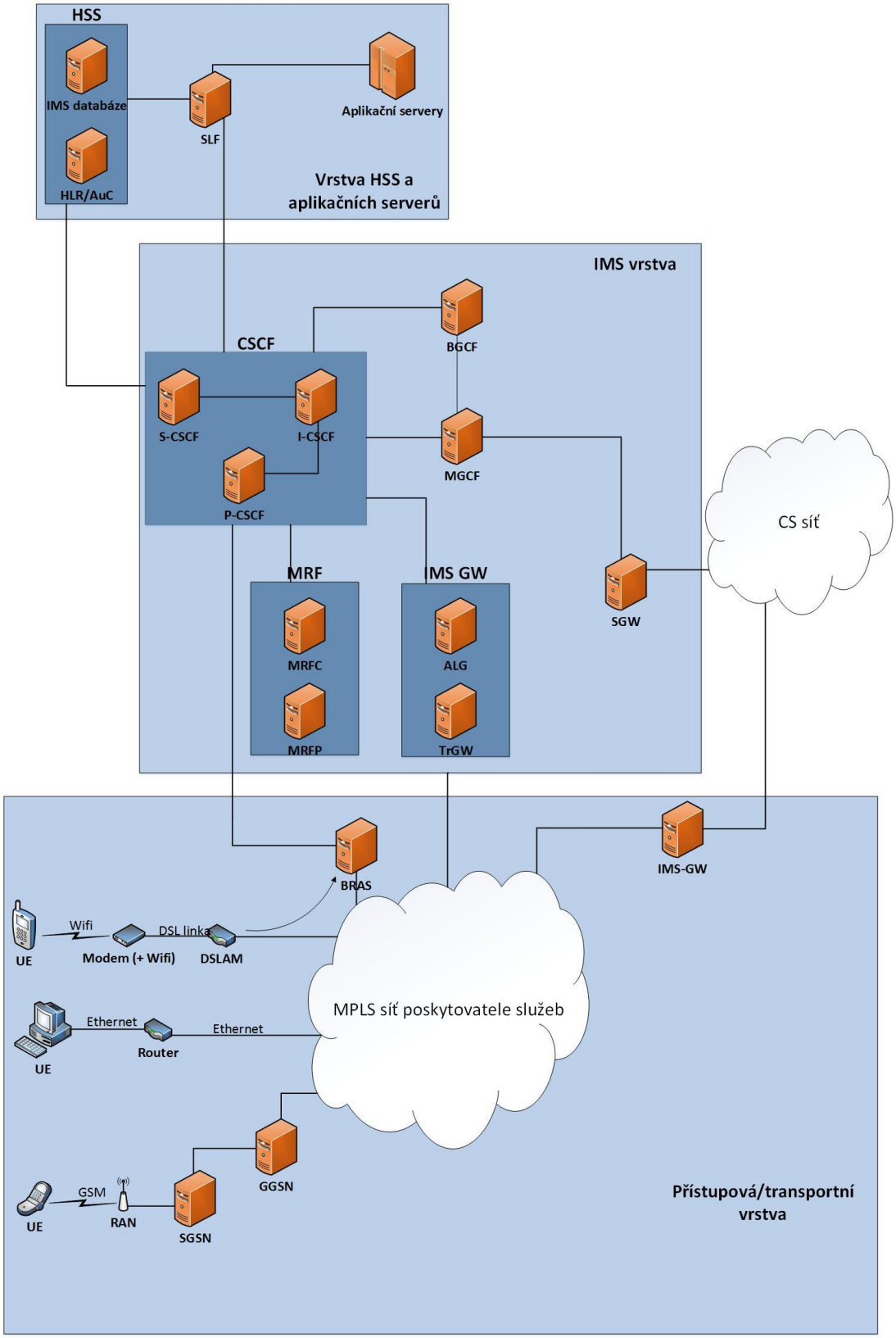
Na Obrázku 1 je zjednodušený diagram IMS sítě. Obsahuje všechny klíčové prvky sítě, o kterých bude řeč v dalších kapitolách.

2.1.1 Transportní vrstva

Transportní vrstva by se dala nazvat přístupovou vrstvou. Zahrnuje všechny přístupové technologie a k nim nezbytné další funkční bloky. Například pro xDSL by šel uvést BRAS (*Broadband Remote Access Server*) server, na kterém končí PPPoE (*Point-to-Point Protocol over Ethernet*) zákazníka. Autorizace a autentizace už však v této vrstvě neprobíhá, jelikož databáze uživatelů patří do vrstvy aplikační. BRAS se tedy s ověřením musí obrátit na databázi uživatelů, místo na RADIUS (*Remote Authentication Dial-In User Service*), jak to funguje dnes.

Do této vrstvy pro představu dále lze zařadit SGSN (*Serving GPRS (General Packet Radio Service*) *Support Node*)/GGSN (*Gateway GPRS Support Node*) mobilních sítí, MPLS (*Multiprotocol Label Switching*) páteřní síť operátora, DSLAM (*Digital Subscriber Line Access Multiplexer*) atd.

Pro moji práci byla však tato vrstva celkem nezajímavá a také nedostupná. Testování jednotlivých přístupových možností nebylo cílem práce a ani v časových možnostech a kromě wifi je i většina technologií pro mě nedostupná – např. testování LTE vzhledem k potřebě licence frekvenčního pásma není bez spolupráce s operátorem možné. V transportní vrstvě však leží už zmíněná největší změna a přínos IMS – je naprosto jedno, jakým způsobem a jakou technologií se uživatel připojil. Jestliže jeho zařízení obsahuje IMS klienta, vyšší vrstvy sítě jeho přístupovou metodu neřeší.



Obrázek 1 - Topologie IMS sítě

2.1.2 IMS vrstva

IMS vrstva je srdcem sítě. Leží v ní tři nejdůležitější součásti ze čtyř (tou čtvrtou je databáze uživatelů). Tyto části se nazývají CSCF (*Call Session Control Function*) a jsou podle funkce tři [5]:

1. P-CSCF – Proxy-Call Session Control Function
2. I-CSCF – Interrogating-Call Session Control Function
3. S-CSCF – Session-Call Session Control Function

Mezi nimi je směrována veškerá signalizace - od prvního požadavku uživatele na registraci a žádost o sestavení hovoru, až po odhlášení uživatele ze sítě. Vše pomocí protokolu SIP. Například při registraci a spojování hovorů musí CSCF často spolupracovat s databází uživatelů, která se označuje HSS (*Home Subscriber Server*).

V této vrstvě jsou dále ještě nejrůznější brány např. pro přechod do sítě jiného operátora nebo pro spojení hovoru do starších typů sítí (GSM (*Global System for Mobile Communications*) nebo PSTN (*Public Switched Telephone Network*)).

Následuje detailní popis funkce jednotlivých bloků IMS vrstvy.

2.1.2.1 P-CSCF

Proxy-CSCF je prvním kontaktním místem každého uživatele, resp. jeho UE. Mezi P-CSCF a UE se tak navazuje veškeré zabezpečení, u IMS to může být typicky IPSec (*IP Security*). P-CSCF musí tedy udržovat veškeré informace o zabezpečení – například SA (*Security Association*) jednotlivých tunelů.

Jelikož je SIP textový protokol a obsahuje velké množství informací v hlavičkách jednotlivých zpráv, využívá se také komprese signalizace – SigComp (*Signalling Compression*), o tuto činnost se také stará P-CSCF. Další úkoly pak zahrnují komunikaci s prvky sítě pro účtování, kdy IMS může posílat a získávat informace do a z přístupové sítě. Jako poslední provádí P-CSCF detekci nouzových spojení – hasiči, policie, ZZS (*Zdravotnická*

záchranná služba), které se pak zpracovávají pochopitelně s prioritou a jiným způsobem než běžné hovory.

Při přihlášení účastníka do sítě je adresa, resp. název P-CSCF jediná nutná věc pro připojení.

2.1.2.2 I-CSCF

Interrogating-CSCF je vnitřním bodem sítě operátora. Pro P-CSCF zjišťuje jména dalších entit, které jsou potřebné pro spojení. Při prvotní registraci tedy kontaktuje pro P-CSCF registr uživatelů (HSS) a zjistí, jaký S-CSCF má uživateli přidělit, v závislosti na potřebách zařízení a sítě.

2.1.2.3 S-CSCF

Session-CSCF je nejdůležitější ze všech tří CSCF. Udržuje veškeré údaje o stavu přihlášení uživatele. Při prvotní registraci kontaktuje HSS, stáhne si uživatelův profil a vyzve UE k ověření identity. Poté dohlíží na veškerou další aktivitu při registraci.

S-CSCF si z HSS stáhne uživatelův profil, který může obsahovat mnoho informací, jak k samotnému uživateli, tak i k jeho zařízení. Například si uživatel u operátora platí pouze audio hovory, případně si může platit konferenční hovory. UE také například nemá displej/kameru a tak není možné poslat uživateli na toto zařízení videohovor atd.

S-CSCF dále kontaktuje všechny potřebné aplikační servery a pokud hovor nepokračuje v IMS síti, předává relaci dalším entitám pro přechod do např. PSTN sítě.

I když S-CSCF zná IP adresu UE, nikdy nedochází k přímému kontaktu – je potřeba signalizaci předat skrze P-CSCF, aby došlo k zabezpečení dat prostřednictvím IPSec tunelu.

2.1.2.4 Síťové funkce

IMS koncept by měl být univerzální sítí. Jelikož není možné přemigrovat všechny uživatele na nová zařízení, která budou schopná se do IMS sítě připojit přes SIP v jeden okamžik

(uživatel si musí koupit nový telefon), IMS síť musí umožnit spojení hovoru mezi uživateli, jejichž zařízení používá CS, či PS sítě se všemi možnými druhy signalizace a zároveň také opačně – ze starších sítí do IMS.

K tomuto účelu slouží hned několik entit. S-CSCF rozhoduje, jestli dojde k předání relace do jiné sítě. Pokud zjistí, že ano, předává relaci na BGCF (*Breakout Gateway Control Function*). Zde musí dojít k několika dalším rozhodnutím. Záleží, jestli je hovor určen do sítě stejného operátora, pouze starší typ sítě. V takovém případě relace putuje od BGCF k MGCF (*Media Gateway Control Function*), kde se musí transformovat signalizace – z protokolu SIP na ISUP (ISDN User Part). MGCF kontroluje další prvek sítě – IMS-MGW. Zde se převede audio mezi protokoly v obou druzích sítě. Pokud hovor přichází z CS sítě, prochází nejprve přes MGCF, kde jsou provedeny výše zmíněné úkony a dále je už SIP signalizace posílána na I-CSCF, který zajistí správně směrování.

Další případ může být předání do IMS sítě jiného operátora, poslední možností je starší typ sítě jiného operátora. V obou případech dochází k předání na BGCF cizí sítě a dále se pokračuje postupem přes MGCF popsaným výše.

2.1.2.5 Podpůrné funkce

IMS v sobě zahrnuje mnoho podpůrných funkcí, jejichž činnost je buď okrajová, nebo v závislosti na typu sítě ani nemusejí být implementovány. Jedná se například o vyhledávání geograficky nejbližších středisek pro tísňová volání, různé funkce pro technické zajištění spojení relací mezi sítěmi dvou operátorů nebo bezpečnostní funkce pro zabezpečení dat v síti operátora.

2.1.3 Servisní/Aplikační vrstva

2.1.3.1 HSS

HSS je hlavní databází sítě. Dá se říci, že v sobě obsahuje všechny prvky pro každou myslitelnou starší technologii. Pro GSM a UMTS (Universal Mobile Telecommunications

System) sítě slouží například jako HLR (*Home Location Register*) a AuC (*Authentication Center*).

HSS obsahuje informace o servisním profilu uživatele, jeho veřejné a privátní identity v IMS. Ví, jaké funkce S-CSCF uživatel potřebuje a při registraci tyto informace předává I-CSCF. Drží také záznamy o roamingu.

Na rozdíl od CSCF (dáno geografickou polohou) entit nemusí být nutně víc HSS v jedné síti, ale vzhledem k redundanci a kapacitě je takový stav žádoucí a pravděpodobný.

2.1.3.2 Servisní funkce – Aplikační servery, MRFC, MRFP

IMS síť sama o sobě nenabízí svým uživatelům příliš. O všechno ostatní se musí postarat především aplikační servery. Ty doplňují funkce jako konferenční hovory, sběr dat o aktuálním stavu uživatelů (služba Presence), která pak využijí třeba komunikátory pro zobrazení, jestli je uživatel „Online“ či „Away“. Aplikační servery jsou umístěny do nejvyšší vrstvy a částečně jsou už mimo IMS síť. Jejich tvůrce může být někdo úplně jiný než výrobce IMS sítě. Nemusí je ani provozovat operátor ve své síti. Výrobce IMS sítě například může poskytnout API (*Application Programming Interface*) operátorovi nebo ho úplně zveřejnit a jakákoliv další firma pak může doplnit svůj aplikační server do sítě a umožnit uživatelům využít dalších služeb. Operátor vše může samozřejmě účtovat podle nejrůznějších kritérií, smluv a dohod.

MRFP (*Media Resource Function Processor*) a MRFC (*Media Resource Function Controller*) jsou spíše podpůrné funkce a umožňují provádět operace s daty uživatelů. Příkladem může být propojení audia a videa od jednotlivých uživatelů do konferenčního hovoru.

2.1.4 Referenční body

Referenční body nejsou žádnou fyzickou součástí sítě IMS. Slouží pro popis komunikace mezi jednotlivými entitami sítě a pro usnadnění popisu, jak probíhají procesy, kudy je vedena signalizace atd.

Referenčních bodů je velké množství. Každý spoj mezi jednotlivými entitami má vlastní označení. Entity však nejsou propojené principem „každý s každým“, což počet referenčních bodů redukuje na rozumnou úroveň. V následujících podkapitolách rozeberu postupně jednotlivé body.

2.1.4.1 Gm

Slouží pro komunikaci mezi CSCF, resp. P-CSCF a UE [6]. Tedy veškerá komunikace, která probíhá mezi sítí IMS a uživatelovým zařízením, musí projít skrz bod Gm.

Jako první je vždy registrace, nebo pokus o registraci. Před registrací navíc musí proběhnout důležité procesy pro sestavení IPSec tunelu, kterým poté proudí už šifrovaná komunikace. Při odhlášení ze sítě samozřejmě tímto bodem jde i veškerá deregistrační signalizace.

Kromě SIP signalizace tímto bodem prochází samozřejmě i hovory, případně i multimediální zprávy.

2.1.4.2 Mw

Referenční bod Mw spojuje dohromady všechny CSCF. Vyměňují se tedy přes něj všechny důležité signalizační zprávy. Stejně jako přes Gm zde najdeme 3 typy provozu:

1. Registrace – při registračním procesu se mezi jednotlivými CSCF vymění spousta signalizace. Téměř každá SIP zpráva prochází přes všechny 3 nebo pouze 2 CSCF. Díky tomu lze poměrně pohodlně sledovat (například programem na analýzu síťového provozu) signalizační proces. Jak ukážu v následujících částech, při budování sítě a odstraňování problémů je tato analýza téměř nejdůležitější.
2. Relace – najdeme zde samozřejmě i nutnou signalizaci pro sestavení relace (hovoru) mezi účastníky. SIP zprávy jsou totožné s těmi, jaké bychom našli v Gm, pouze P-CSCF představuje zdroj těchto zpráv. Příslušná signalizace zde proudí i při ukončení hovoru.

3. Transakce – typicky textové zprávy, jejich posílání IMS umožňuje, případně potvrzení SIP protokolu.

2.1.4.3 ISC

ISC je jedním ze dvou bodů, kterými IMS komunikuje s aplikačními servery. Tedy konkrétně pro ISC je to S-CSCF, které komunikuje s AS. Například při registraci nebo jiném požadavku je zřejmé (podle dat, které si S-CSCF stáhne z HSS), že by měl být kontaktován aplikační server, a tak ho S-CSCF i informuje. Aplikační server se pak rozhodne, co dál bude s přijatou zprávou nebo daty dělat. Opačně i aplikační server může přes referenční bod kontaktovat S-CSCF – například v případě, že potřebuje uživateli doručit textovou zprávu nebo získat informace o jeho momentálním statusu (jestli je uživatel „online“, „away“ a tak podobně).

2.1.4.4 Ma

V Release 7 byl představen lepší způsob směrování některých procesů, které nepotřebují interakci s S-CSCF, a tedy šetří jeho zdroje a komunikují přímo s aplikačním serverem. Právě tato komunikace probíhá přes *Ma* referenční bod.

2.1.4.5 Cx

Cx je určen pro výměnu dat mezi CSCF a HSS. Na rozdíl od předchozích komunikací není ve formě SIP zpráv, ale pomocí protokolu Diameter. Data se zde vyměňují v poměrně specifické podobě – jednotlivé entity mohou od HSS vyžadovat určitá data při procedurách s přesně danou formou. Například UAR (*User Authorization Request*) použije I-CSCF při registraci uživatele k tomu, aby byl zjištěn správný S-CSCF. Pro stáhnutí uživatelova profilu z HSS pak S-CSCF používá SAR (*Server Assignment Request*).

2.1.4.6 Dx

V síti s více HSS není při registraci zřejmé pro I-CSCF a ani pro S-CSCF, která databáze by se měla použít. K tomu slouží SLF (*Subscription Locator Function*). Dx je referenční bod, přes

který komunikují CSCF a SLF. Jak napovídá z názvu referenčního bodu písmeno „x“, jedná se stejně jako u Cx o referenční bod, který využívá protokol Diameter.

Když chce I-CSCF nebo S-CSCF zjistit adresu příslušného HSS, pošlou totožný požadavek, jaký by poslali na HSS přes Cx, přes Dx na SLF. Odpovědí jim je adresa správného HSS, kterému pak můžou už směrovat přes Cx opět zmíněný požadavek.

2.1.4.7 Sh

SH referenční bod je určen pro komunikaci mezi aplikačním serverem (AS) a HSS. Jelikož HSS obsahuje data, která některé aplikační servery mohou nutně potřebovat, tento referenční bod takovou komunikaci umožňuje. Komunikační protokol je opět Diameter. Aby se předešlo neoprávněným vstupům do dat uživatele, udržuje HSS seznam povolených aplikačních serverů, které mohou od něj žádat informace.

2.1.4.8 Dh

Stejně jako u referenčního bodu Dx, pokud existuje v síti více HSS a aplikační server potřebuje pomocí referenčního bodu získat nějaká data, nemůže vědět, na který HSS se obrátit. Využije tedy stejný postup jako AS (Application Server) a nejdřív od SLF zjistí, který HSS má využít. Protokol je opět Diameter. Postup zjištění je podobný jako u Dx.

2.1.4.9 Mi

Mi referenční bod je určen pro komunikaci mezi S-CSCF a BGCF v případě, že je potřeba relaci směrovat do CS sítě. Použitým protokolem je SIP.

2.1.4.10 Mj

Mj referenční bod BGCF využije v případě, že je relaci potřeba směrovat do CS sítě stejného operátora. Protokolem je opět SIP a relace je směrována na MGCF pro konverzi signalizace.

2.1.4.11 Mk

Mk je v podstatě opakem Mi – pokud je potřeba relaci směrovat do CS sítě, ale v cizí síti, využívá IMS, resp. BGCF, Mk referenční bod, kterým komunikuje s BGCF cizího operátora. Protokol je opět SIP.

2.1.4.12 Mg

Tento referenční bod využívá MGCF v případě, že jde o příchozí relaci z CS sítě. Spojí se jeho pomocí s I-CSCF (po konverzi z ISUP na SIP) a předá potřebné SIP zprávy.

2.1.4.13 Mr

V případě, že S-CSCF potřebuje uživateli přehrát nějaké zvuky, kontaktuje MRFC. Tato komunikace využívá Mr referenční bod. Použitým protokolem je SIP.

2.1.4.14 Mp

Mp referenční bod je velice blízký Mr. MRFC kontroluje hraní jednotlivých tónů, zvuků či oznámení, ale entitou, která toto provádí je MRFP. Mp je komunikační rozhraní právě mezi těmito entitami. Oproti jiným referenčním bodům se zde podle ITU-T (International Telecommunication Union-Telecommunication) využívá H.248.

2.1.4.15 Gx

Referenční bod Gx spojuje PCRF (*Policy and Charging Rules Function*) a přístupovou bránu technologie, přes kterou se uživatel připojil. Díky tomu může mít operátor kontrolu nad provozem, který od uživatele přichází a může provádět různé operace.

Těmi jsou například:

- Ovládat QoS (Quality of Service) pro jednotlivé hovory/uživatele/aplikace.
- Nastavovat bezpečnostní pravidla – například na Firewallu.
- Aktivovat účtování – jestli jde o tzv. „offline“ nebo „online“ účtování.
- Měřit různé parametry – délku hovoru atd.
- Monitorovat a reportovat ukončení či přerušení hovoru.

Použitým protokolem je Diameter.

2.1.4.16 Rx

Plní v podstatě stejnou činnost jako Gx, resp. přenáší se zde téměř ta samá data se stejným účelem. Hlavním rozdílem je, že Rx se nachází mezi P-CSCF a PCRF. Pokud P-CSCF detekuje jakoukoliv SIP zprávu, která obsahuje SDP (*Session Description Protocol*), informuje o tom PCRF. Ta má tak všechny aktuální informace pro například účtování.

2.1.4.17 Další referenční body

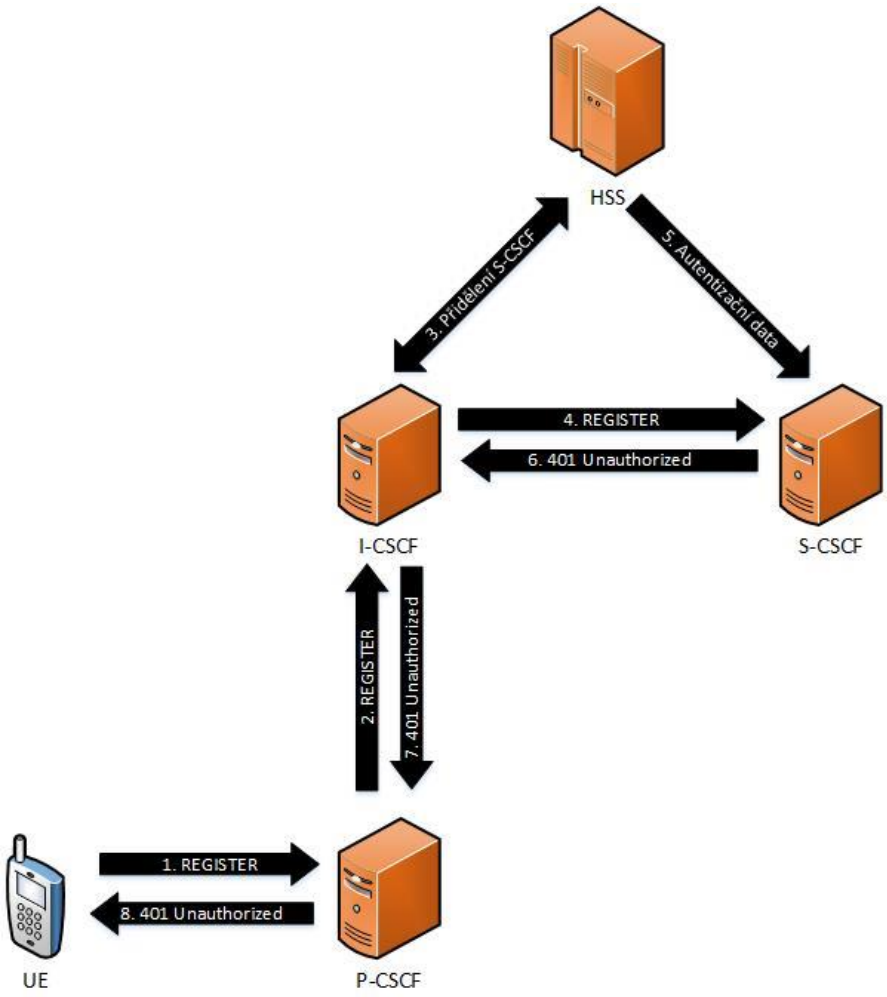
Existuje mnoho dalších referenčních bodů, ale jejich funkce je buď okrajová (spojují entity, které jsou spíše rozšiřující pro IMS) anebo o nich bude více řeč dále.

2.2 Registrace

Jak již bylo řečeno, UE s IMS sítí vždy komunikuje přes P-CSCF. Není tomu jinak ani u registrace. Zařízení odešle *REGISTER* na P-CSCF, které požadavek přepošle na I-CSCF a to zjistí od HSS, jaké má využít S-CSCF. Paralelně HSS odešle S-CSCF autentizační data pro daného uživatele. S-CSCF zjistí, že uživatel není ověřen a zašle mu ověřovací data. (tzv. challenge). Tato část registrace je na Obrázku 2.

V druhé fázi UE spočte odpověď svým algoritmem a znovu odešle *REGISTER*. I-CSCF znovu zjistí, jaké S-CSCF má použít, S-CSCF znovu ověří uživatele, a protože vše souhlasí, stáhne si

z HSS celý uživatelům profil. Pak od S-CSCF přes I-CSCF až k UE projde *200 OK*. Obnovení registrace je potřeba provádět nejpozději s vypršením limitu pro registraci.

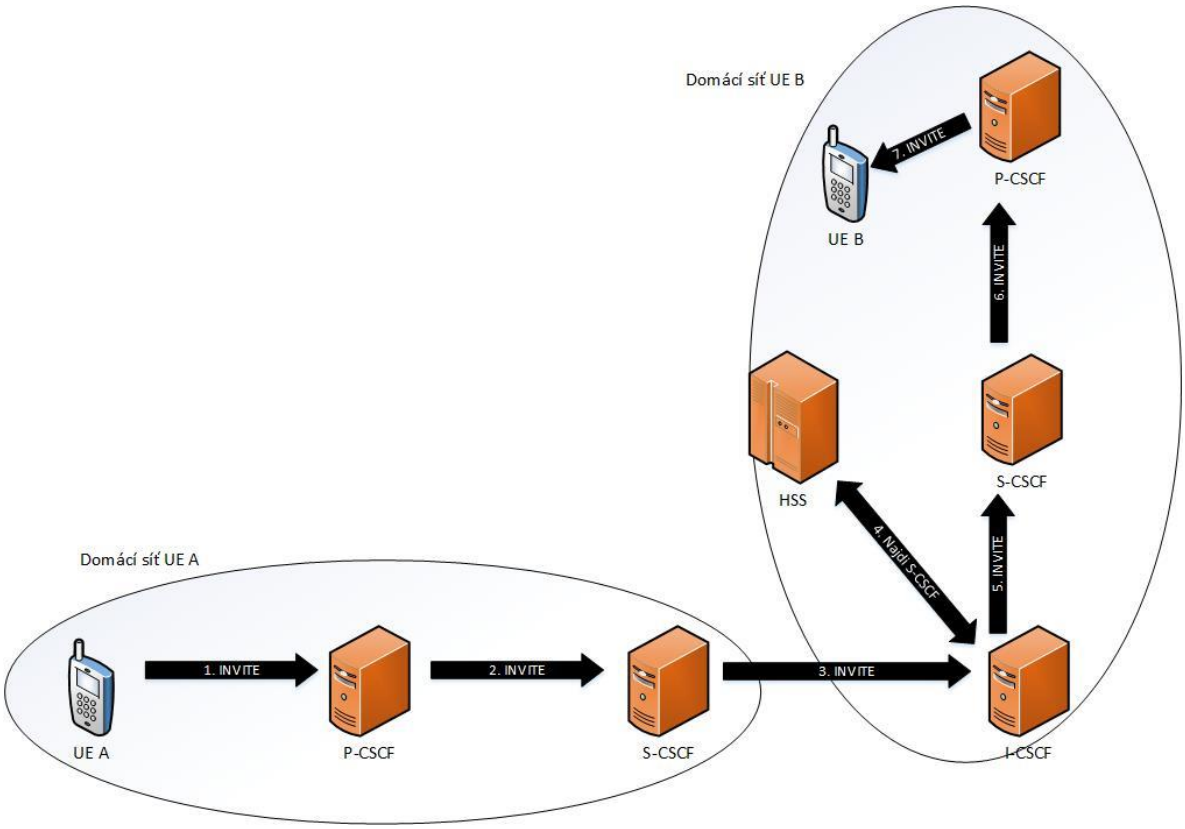


Obrázek 2 - První fáze registrace

2.3 Sestavení relace

Na rozdíl od protokolu SIP, kde koncová zařízení mohou komunikovat napřímo, v IMS je tato procedura o poznání složitější. Pokud chce zařízení UE A komunikovat s UE B, které je navíc v jiné IMS síti, musí INVITE projít přes několik prvků sítě. Vstupní a výstupní body sítě jsou vždy P-CSCF. Od prvního P-CSCF tedy INVITE putuje k S-CSCF (hledat S-CSCF a ověřovat uživatele už není třeba, to se dělo při registraci). Dále se INVITE předá k I-CSCF cílové sítě, která však už musí zjistit z HSS, které S-CSCF uživateli slouží. Přes S-CSCF INVITE

putuje k P-CSCF a pak k UE B. Obrázek 3 tuto proceduru ilustruje. Stejným způsobem pak probíhají i další zprávy, které SIP vyžaduje k sestavení relace, podrobnější popis je v kapitole 3.4.



Obrázek 3 - Sestavení relace v IMS

2.4 Identity

V IMS jsou pro uživatele používány dva různé druhy identit.

- Veřejná – slouží pro identifikaci uživatele pro vnější svět, může to být SIP URI, nebo také identita převoditelná na klasické telefonní číslo
- Privátní – používá se pouze pro záznam a registraci uživatele v jeho síti a je uživateli vydána s počátkem užívání služby a s účtem u operátora je pevně svázána.

2.4.1 Veřejná identita

Hlavním znakem veřejné identity uživatele je to, že jich může být víc. Uživatel může mít několik pracovních a několik soukromých identifikátorů, které jsou připojeny na jeho účet. Všechny musí splňovat několik pravidel.

- Mít tvar buď podle SIP URI (Session Initiation Protocol Uniform Resource Identifier), nebo formát telefonního čísla.
- Jedna veřejná identita musí být uložena na SIM kartě a UE ji nemůže měnit.
- Síť neověřuje veřejné identity při registraci.

2.4.2 Privátní identita

Jak jsem již uvedl, privátní identita je natrvalo spojena s uživatelovou smlouvou u operátora. Nemá identifikovat člověka, ale účet. Proto je hlavně určena pro autorizaci a autentizaci uživatele. Následuje několik faktů o privátní identitě.

- Má tvar NAI (*Network Access Identifier*) specifikovaný v RFC 4282 [7].
- Bude vždy obsažena ve všech registracích uživatele a bude se ověřovat při každé registraci, obnovení registrace i jejím ukončení.
- Je uložena v ISIM (*IMS Identity Module*) a UE ji nemůže měnit.
- Mohou podle ní být zákazníkovi fakturovány služby.

2.4.3 Identita zařízení

V IMS je možné jednoznačně identifikovat uživatele, ale také zařízení. Několik použití se přímo nabízí. Jedním z nich je, když uživatel sám označí některé zařízení jako nežádoucí pro hovor – například odejde od pracovního stolu a nepřeje si, aby mu vyzváněl stolní telefon, ale nechce ho odhlašovat ze sítě. Označí ho tedy nějakým způsobem a síť na telefon nebude směrovat hovory.

Další možné využití vyplyne z faktu, že ne všechna zařízení jsou schopna zpracovat jakýkoliv druh hovoru – například obyčejný IP telefon, nebo starší mobilní telefon se

s videohovorem těžko vypořádá. Proto v případě, že má uživatel zaregistrovaných v jednu chvíli více zařízení, síť zná jejich schopnosti a směruje určité typy relací pouze na vhodná z nich.

Jako poslední příklad lze uvést předání hovoru z jednoho zařízení na druhé – musí existovat jejich identifikátor. Aby toto bylo možné, IMS využívá identifikátor zařízení nazvaný GRUU (*Globally Routable User Agent URI*). Existují dvě varianty GRUU:

- Veřejná – spojuje zařízení s identitou uživatele do té doby, dokud uživatel zařízení vlastní a umožňuje tak kdykoliv kontaktovat konkrétní zařízení.
- Dočasná – neobsahuje veřejnou identitu uživatele a ta tak zůstává skrytá. Dočasné GRUU se vytvoří, když se nové zařízení přihlásí do sítě.

2.5 IMS služby

2.5.1 Presence

Služba Presence se zdá být skoro jako samozřejmost nebo možná jako okrajová záležitost. Ve skutečnosti podobné funkce využíváme v podstatě denně. Ať je to Facebook, který dává vědět, jestli je uživatel online, případně před jakou dobou byl. Nebo například firemní komunikátor, kde uživatelé mohou dát vědět nastavením svého stavu, že jsou zrovna na schůzce a nechtějí být rušeni. Podobné komunikátory jsou využívány i v soukromém životě – přibývá i třeba nastavení nálady atp. Všechny tyto funkce SIP nabízí.

Pro Presence slouží vlastní Presence server, kde jsou informace o stavu uživatelů uloženy. Pomocí zprávy *SUBSCRIBE* se zařízení může přihlásit k odběru informací o daném uživateli a je pak v intervalech vyrozuměno zprávou *NOTIFY*. Pro potvrzení zprávy *SUBSCRIBE* se použije *202 Accepted*.

3 SIP

Protokol SIP je popsán v RFC 3261 [8] a je určený k sestavení relací – jak je zřejmé z jeho názvu. Tyto relace však i běžně ruší. Pro sestavení relace umožňuje výměnu všech nutných informací.

SIP je typickým představitelem protokolu aplikační vrstvy OSI modelu (*Open Systems Interconnection).* Model OSI zde nebude rozebírán a tak jen uvedu, že na aplikační vrstvě se SIP nachází ještě například s DNS (*Domain Name System*) či RTP (*Real-time Transport Protocol*), jež IMS také velmi významně využívá. SIP sám o sobě nepřenáší žádný hlas, pakety nebo segmenty, které by hlas obsahovaly. Na přenosu multimediálních dat spolupracuje právě s RTP.

Kromě RTP, SIP využívá hojně i SDP, které se nachází vždy v těle SIP zprávy a dojednává kodeky pro následnou audio nebo i video komunikaci [9].

Pokud je třeba SIP zabezpečit, lze využít IPSec či TLS (*Transport Layer Security*), pro zabezpečený přenos hlasu lze využít SRTP (*Secure Real-time Transport Protocol*).

Ačkoliv SIP je jen jednou z několika součástí potřebnou pro provedení hovoru, pro IMS představuje SIP zásadní protokol, bez kterého by vůbec nemohla síť fungovat. Dalo by se říci, že SIP je pro IMS nejdůležitější součástí vůbec. Veškerá komunikace mezi všemi entitami, s výjimkou HSS a SLF, které komunikují pomoci Diameteru, je totiž prováděna s pomocí SIPu.

3.1 Stručná charakteristika

SIP má za účel sestavit relaci. Pod tou si nejčastěji lze představit sestavení běžného hovoru mezi dvěma účastníky. Stejně jako HTTP (*Hypertext Transfer Protocol*), je SIP „textový“ protokol – všechny jeho zprávy lze po zachycení běžně číst a není potřeba žádná interpretace. Na rozdíl od jiných protokolů aplikační vrstvy nelze jednoduše říci, jestli pro

svůj přenos SIP využívá na transportní vrstvě protokol TCP (*Transmission Control Protocol*) nebo UDP (*User Datagram Protocol*) – využívá totiž obojí, nejčastěji s portem 5060. Aby byl popis vrstev OSI modelu kompletní, na třetí vrstvě je protokol dnes snad už vždy IP a na nižších vrstvách záleží na použitém hardwaru, i když nejčastěji na druhé vrstvě dnes jde o Ethernet, případně PPPoE.

3.2 SIP požadavky

Stručná charakteristika SIP požadavků [10]:

Požadavek	**Popis**
INVITE	První výzva sestavení relace
ACK	Finální potvrzení odpovědí (další kapitola)
BYE	Ukončení relace
CANCEL	Zrušení probíhajících požadavků a pokusů o sestavení relace
OPTIONS	Zjištění schopností a stavu SIP zařízení
REGISTER	Zařízení informuje síť o své současné identitě
PRACK	Potvrzení odpovědí z kategorie 1xx (další kapitola)
SUBSCRIBE	Přihlášení se k odběru notifikací (obvykle na server)
NOTIFY	Sdělení informací druhé straně o problému/události
PUBLISH	Odeslání informací na server
INFO	Zaslání nových informací druhé straně v průběhu hovoru
REFER	Požádání jiného klienta o využití URL nebo URI
MESSAGE	Přenos zprávy prostřednictvím SIP
UPDATE	Změna parametrů relace v jejím průběhu

Tabulka 1 - SIP požadavky

3.3 SIP odpovědi

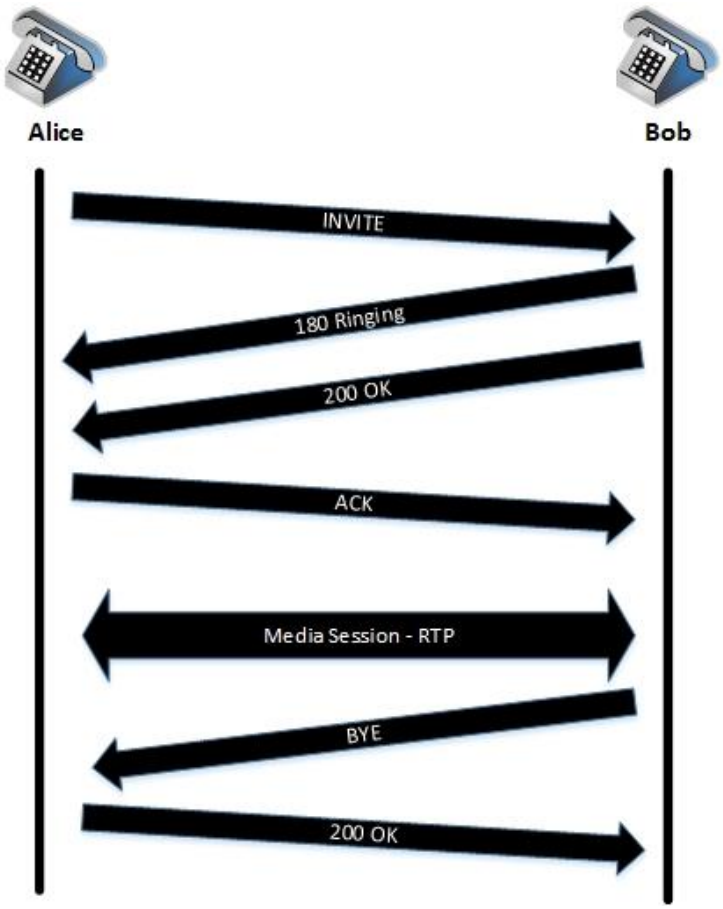
Odpověď, kterou může zařízení pracující se SIP protokolem zaslat na požadavek, patří do jedné z 6 kategorií.

Třída odpovědi	Kategorie	Popis
1xx	Informační	Informace o stavu relace před jejím sestavením nebo odmítnutím.
2xx	Úspěch	Potvrzení o úspěšném vyřízení požadavku.
3xx	Přesměrování	Zpráva o přesměrování cíle. Obsahuje novou adresu cíle.
4xx	Chyba na straně uživatele	Požadavek selhal kvůli chybě na straně uživatele. Uživatel by měl upravit požadavek před dalším pokusem.
5xx	Chyba na straně serveru	Požadavek selhal kvůli chybě serveru, uživatel může zkusit jiný server.
6xx	Všeobecná chyba	Požadavek by neměl být zkoušen znovu zde, ani na jiném serveru.

Tabulka 2 - SIP odpovědi

3.4 Sestavení relace

Díky tomu, že SIP je textově orientovaný protokol, lze si přehledně zobrazit průběh sestavení relace s názvy zpráv tak, jak je možné vidět přímo v softwaru pro analýzu provozu. Celý proces, jak jde signalizace po sobě, lze vidět na Obrázku 4, s nejčastěji používanými uživateli Alice a Bob.



Obrázek 4 - Sestaveni SIP relace

Jako první výzvu pošle ten, kdo chce zahájit hovor zprávu *INVITE* [11]. V této zprávě je uvedeno několik zásadních údajů, mezi nimi jde především o:

- O jakou zprávu se jedná (*INVITE*, *OK*, *BYE* atd.).
- Komu je určena a od koho pochází (využívají se SIP-URI, tedy identifikátory, které je potřeba přeložit pomocí DNS).
- Max-forwards – obdoba TTL (Time-To-Live) z IP paketu, která má zabránit nekonečnému obíhání dat v síti v nekonečných smyčkách [12].
- Označení relace, aby bylo možné ji odlišit od jiných.

V obsahu SIP zprávy pak dále může být protokol SDP, který specifikuje multimediální informace – jestli se bude přenášet jen audio nebo i video, jaké kodeky lze využít atd. Příklad, jak může hlavička zprávy *INVITE* vypadat, je na Obrázku 5.

Obrázek 5 - Příklad zprávy INVITE

Odpovědí na *INVITE* je *180 Ringing*. Jako každá odpověď protokolu SIP patří to jedné z kategorií charakterizovanými podle první číslice, tedy 1xx (dále 2xx, 3xx atd.). 1xx patří mezi informační zprávy SIPu a informuje zdroj o vývoji v sestavování relace. Mnoho SIP odpovědí je shodných se stejnými čísly u HTTP. Pro zařízení pracující se SIPem není ani tak důležité, jak se daná odpověď nazývá, zásadní je její číslo. Například kategorie 4xx označuje chybu uživatele, z HTTP se používá i u SIPu odpověď 404. U SIPu jde o odpověď v případě, že cílový uživatel nebyl nalezen a není zřejmé, kam zprávu poslat. Struktura *180 Ringing* je velmi podobná *INVITE*, ale musí být jasné, na jakou původní zprávu odpovídá. Jak je jinak z názvu zprávy celkem zřejmé, bývá odeslána cílovou stanicí ve chvíli, kdy začne vyzvánět.

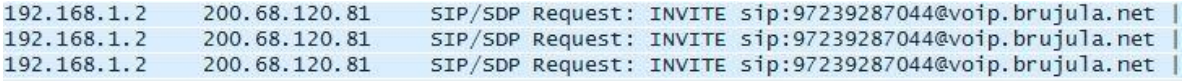
Po zvednutí sluchátka uživatelem je zapotřebí dát zdrojové stanici vědět, že lze začít přenášet audio, případně video, data samotného hovoru. To má na starosti odpověď z kategorie 2xx, které označují úspěšný proces, vyhovění požadavku atp. Kromě samotného zvednutí telefonu 200 OK navíc dá vědět zdrojové stanici, že je možné hovor realizovat za použití určitých kodeků, na nichž se obě stanice domluvily (protokolem SDP v těle zpráv SIP).

Následuje přenos multimediálních dat, audia, videa, případně i textových zpráv a souborů. Pro tento účel se obvykle využije už zmíněný protokol RTP.

Hovor ukončuje jedna ze stran zprávou BYE, kterou druhá strana potvrdí 200 OK.

V tomto případě SIP využívá protokol UDP. Ten je známý tím, že negarantuje doručení do cíle. Pro SIP a zvlášť pro následný přenos hlasu je však důležitý co nejmenší „overhead“, který by ubral z přenosové kapacity a mohl zvýšit odezvu. Kvůli nezaručenému doručení signalizace posílá zdroj někdy zprávy i vícekrát, aby zvýšil pravděpodobnost doručení. Jak tento mechanismus vypadá v protokolovém analyzátoru, je vidět na Obrázku 6. Retransmisi zpráv se předchází potvrzením zprávy zpět zdroji. Problém může nastat na zahlcených a pomalých připojeních – UDP pakety se nedostanou k cíli, od cíle nepřijde potvrzení, zdroj tedy posílá další a další a přetížení připojení se ještě více zhoršuje.

SIP navíc nepodporuje fragmentaci vlastních zpráv (větší zprávy s mnoha hlavičkami mohou mít větší velikost než je MTU (*Maximum Transmission Unit*). Ta se tedy musí provádět na transportní vrstvě. Jak bylo řečeno, SIP častěji používá UDP, ale ty není při fragmentaci možné seřadit, a tak je lepší pro fragmentované SIP zprávy použít TCP.



Obrázek 6 - Opakování *INVITE* zdrojem

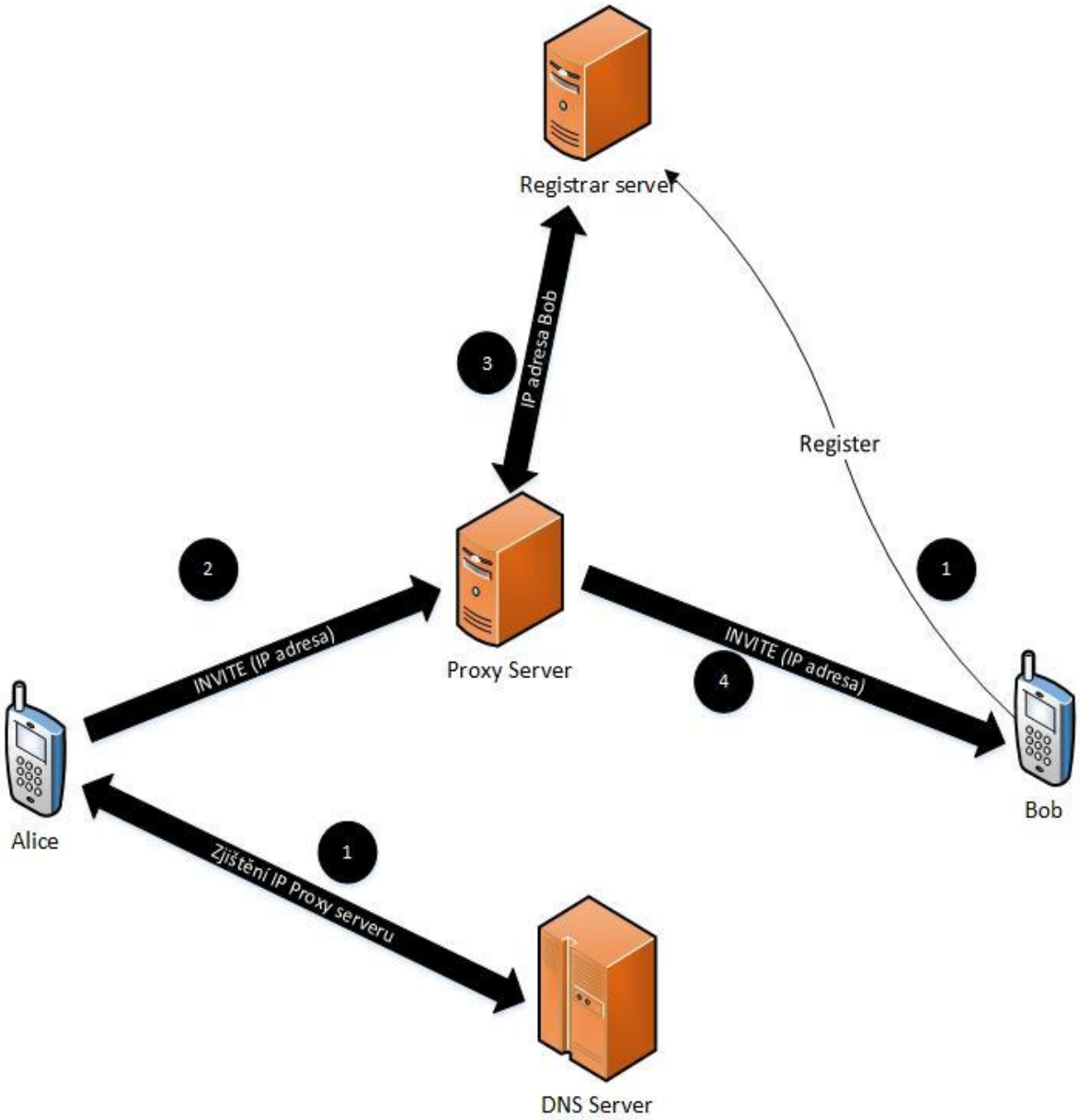
3.5 Identita uživatele

SIP, resp. SIP-URI dovoluje mnohem větší flexibilitu pro uživatele než dosavadní telefonní čísla. Telefonní číslo, na které je zatím stále ještě zvyklá většina lidí, je vázáno na fyzický telefonní přístroj, resp. u mobilních telefonů na SIM kartu. To nedovoluje nijak manipulovat s přihlášením do sítě bez toho, aby se musela SIM karta vyjmout, případně u nepřenosných telefonů řešit změnu čísla s operátorem (tzv. pevné linky v klasickém významu už však příliš rozšířené nejsou). SIP přináší tu výhodu, že se uživatel pod svojí identitou může přihlásit kdekoliv, z jakéhokoliv přístroje a ani to nemusí být telefon, ale například počítač – může využít sluchátka s mikrofonem a mít volné ruce na další práci. Pro častější, především pracovní hovory (call centra atp.) je to takřka nezbytnost. To však dovolovaly do jisté míry i tzv. „hands-free“ sady pro mobilní i klasické telefony. Další a hlavní výhodou je, že uživatel může mít hovor směrován domů, do práce, do auta a

pro volajícího jde stále o tu samou identitu, vystupující pod jménem podobným emailové adrese, která nemusí obsahovat pouze čísla, a tak je i snadněji zapamatovatelná. Tento formát adresy se jmenuje SIP-URI.

Ve světě IP však žádná adresa jako SIP-URI neexistuje a pakety musí mít cílovou a zdrojovou IP adresu. Zdrojovou IP pochopitelně volající do paketu doplní snadno, ale nějakým způsobem musí zjistit cílovou. To však u SIP-URI nelze přidělit staticky (lze, ale cílem SIP je, aby to tak nebylo). Musí tedy existovat nějaká entita, která zná uživatelovu IP, která se navíc může podle lokality uživatele měnit. Tou entitou je SIP proxy server.

Proxy server však sám o sobě neví, jestli je uživatel právě přihlášen a pod jakou IP adresou. Aby to zjistil, musí spolupracovat s registračním serverem, označovaným jako SIP Registrar. Tam se uživatel při přihlášení do sítě zaregistruje, tím pádem registrar zjistí jeho IP adresu. Pokud pak volající chce danému uživateli zavolat, neposílá svůj *INVITE* volanému (neví kam), ale pošle ho proxy serveru. Ten zjistí od registrar serveru, jestli je uživatel přihlášen a pod jakou IP adresou a tam mu *INVITE* přepošle. Přiřazení IP adresy k SIP-URI probíhá pomocí DNS. Proxy server může mít veřejnou IP adresu (pokud se jedná o veřejnou síť, tak musí), případně u pracovních sítí může být přístupný pouze z vnitřní firemní sítě (uživatel, který chce pracovat třeba z domova, se nejdříve do firemní sítě přihlásí přes zabezpečenou VPN (*Virtual Private Network*)). Celý proces je na Obrázku 7.

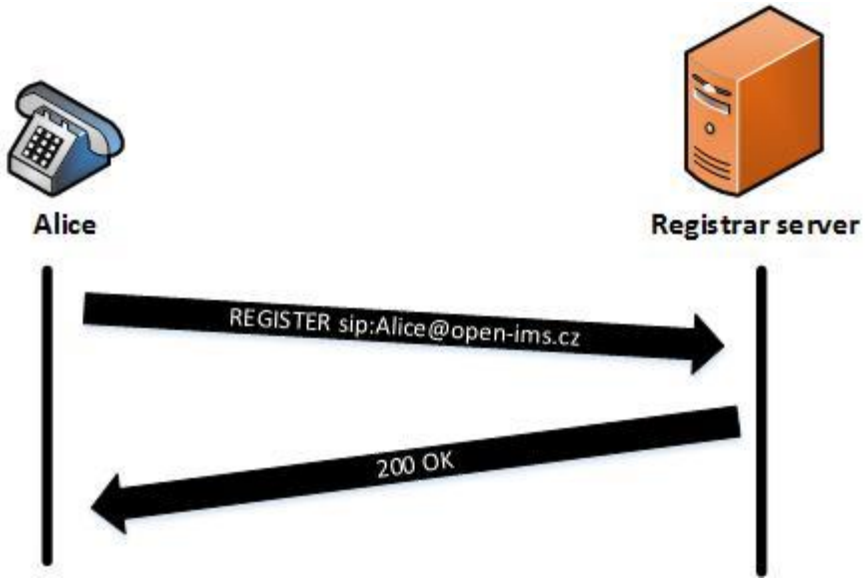


Obrázek 7 - Spojení hovoru při využití SIP-URI

3.6 Proces registrace

V předchozí kapitole a Obrázku 7 byl zmíněn dosud nevysvětlený postup – registrace. Jde o způsob, jak dát vědět své síti, že se tato identita nachází za určitou IP adresou, a také že je možné směrovat na tuto IP adresu hovory, protože uživatel je připraven na ně reagovat.

Oproti sestavení relace je proces registrace výrazně jednodušší, viz Obrázek 8.



Obrázek 8 - Průběh registrace

Zpráva *REGISTER* (Obrázek 9) obsahuje především ve hlavičce „To:“ část, kde je obsažena právě identita, resp. SIP-URI registrujícího se uživatele. V klasické telefonní síti tzv. pevných linek nemá tento proces obdobu, ale je velmi podobný procesu, který proběhne při zapnutí mobilního telefonu a jeho registraci do sítě. Jednotlivé prvky sítě však nevykonávají úplně totožné funkce, a tak je lepší je nepřirovnávat.

Obrázek 9 - Zpráva *REGISTER* ve Wiresharku

Zařízení se neregistruje na neurčitou dobu. V potvrzovací zprávě *200 OK* je pole „expires“, které udává, za jakou dobu se musí registrace obnovit. Pokud si zařízení, resp. jeho uživatel přeje zrušit registraci, pošle *REGISTER* znovu a pole „expires“ nastaví na 0.

3.7 SIP entity

O několika možných typech klientů a serverů už byla řeč v kapitole 3.3. Existuje jich však větší množství.

3.7.1 User Agent – UA

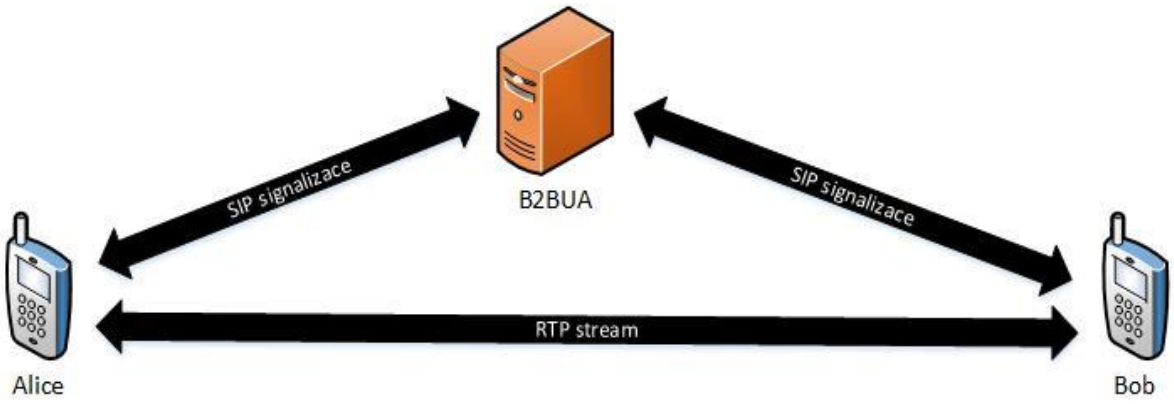
Podmínkou pro SIP UA je, aby mohl sestavovat relace s dalšími UA. Nejčastěji jde o klasický telefon, který využívá přímo člověk, ale může jít i o jiné zařízení, které pro jinou službu zprostředkuje protokol SIP a jeho funkce.

3.7.2 Back-to-Back User Agent – B2BUA

B2BUA ukončuje relace od obou klientů. Chová se tedy jako prostředník. To má několik efektů a využívá se to k různým účelům.

První vlastnost, které se využívá, je to, že klienti nemusí tušit, s kým se baví na druhé straně (aspoň podle SIP to zjistit nemohou). Server tedy slouží jako anonymizér. Další využití vyplývá z jiného faktu – často se za komunikační službu jako je telefon platí. Kvůli účtování musí mít operátor jasný přehled o tom, jak dlouho hovor trval, s kým proběhl,

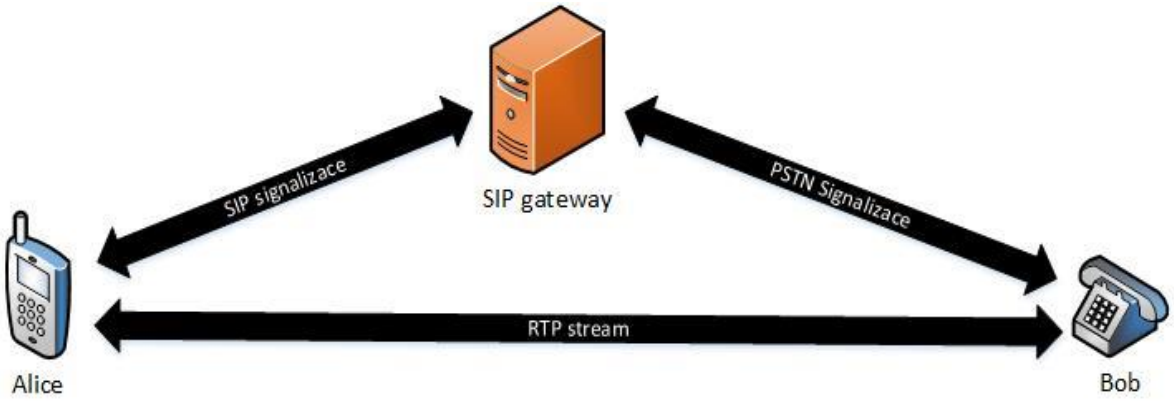
jestli využíval audio, video, konferenci a mnoho dalších služeb – ty pak zákazníkovi může fakturovat. Díky B2BUA a tomu, že veškerá signalizace vede přes tento server, ví operátor tyto detaily o hovoru přesně. Na Obrázku 10 je vidět průběh komunikace.



Obrázek 10 - B2BUA

3.7.3 SIP gateway

V kapitole 2.1.2.4 jsem popisoval přechod hovoru z IMS do CS sítě. U IMS se o to stará několik různých entit (největší podoba by se dala nalézt u MGCF), SIP využívá SIP gateway. Konverzi signalizace provádí tím způsobem, že jednu relaci u sebe ukončí a dále do jiné sítě navazuje další. Pokud se nejedná o přechod ze SIP do PSTN sítě, ale například o konverzi na H.323 kodek, může SIP gateway nechat RTP stream mezi oběma UA bez zásahu a starat se pouze o transformaci signalizace. Použití SIP gateway je na Obrázku 11.



Obrázek 11 - SIP Proxy

3.7.4 Proxy server

Činnost proxy serveru byla částečně popsána v kapitole 3.3. Hlavním rozdílem oproti B2BUA je rozsah, ve kterém oba mohou modifikovat SIP zprávu. Zatímco B2BUA může měnit prakticky cokoliv a zastupovat koncovou stanici, Proxy server je v možnosti modifikace omezen podle RFC 3261 – neměl by modifikovat ani mazat jakékoliv části SIP hlavičky. Díky tomu zachovává možnost stanic komunikovat přímo spolu, bez prostředníka a k uskutečnění komunikace případně jen pomoci poskytnutím nebo dohledáním potřebných dat – dohledání místa uživatele, IP atd.

Proxy server může být:

- stateless – na základě obsahu zprávy ji zpracuje a nic si neukládá, nepamatuje.
- stateful – pamatuje si jednotlivé SIP dialogy a může se snažit zvýšit spolehlivost tím, že znovu pošle za uživatele některé requesty. Může i zvýšit spolehlivost sítě tím, že po uživatelích vyžaduje autentizaci.

Speciální příklad Proxy je tzv. Forking Proxy. Tento efekt nastává v případě, že proxy zjistí, že uživatel je registrovaný na více místech. Proto pošle obdržený *INVITE* na více zařízení a podle toho, jaké přijdou zpět odpovědi (*200 OK* nebo *404 Not Found*), tak postupně maže ze své paměti jednotlivé dialogy. Činnost stateful proxy je na Obrázku 12.

3.7.5 Redirect Server

Redirect slouží, jak název napovídá, k přesměrování požadavků jinam. Využívá k tomu 5 odpovědí z kategorie 3xx. Například na *INVITE* může odpovědět pomocí *302 Moved Temporarily* se správnou kontaktní adresou obsaženou v hlavičce.

3.7.6 Registrační server

Jiným označením jde o SIP registrar. Byl opět zmíněn v kapitole 3.3. Ve své ryzí podobě akceptuje pouze *REGISTER*, na všechny jiné požadavky odpoví *501 Not Implemented*. Registrační server má za úkol udržovat současnou polohu všech momentálně přihlášených uživatelů.

3.8 NAT, TURN a STUN server

SIP je tzv. „end-to-end“ protokol. Aby dvě strany mohly komunikovat, musí se nějakým způsobem spolu spojit. Tomu v dnešním světě vyčerpaných veřejných adres, a privátních adres z RFC 1918, brání NAT (*Network Address Translation*).

Jak jsem již uvedl, SIP je podobný HTTP – protokolu, který nemá velké problémy překonat NAT. Rozdíl je však v tom, že v obvyklém případě je u HTTP vždy server s veřejnou IP adresou a klient s tou privátní (typickým případem jsou domácí sítě a malé podniky). Klient jednoduše pošle požadavek na server a NAT si během spojení udržuje přehled o otevřených zdrojových IP adresách, portech atd., aby mohl příchozí data směrovat správné stanici. Bez otevřeného spojení v paměti NAT však nelze stanici s privátní adresou posílat žádná data.

V případě telefonní stanice však nelze uživateli s privátní adresou určit, že může telefonovat pouze jiným lidem s veřejnou IP adresou, ale nikdo se mu nedovolá.

Proto je vypořádání se s NAT velký problém a zároveň velké téma pro SIP. Využívá se celá řada opatření, jak se s překladem adres vypořádat, jedná se však u všech pouze o řešení problému, který byl vytvořen jiným dočasným řešením. Uvedu pouze několik z nich:

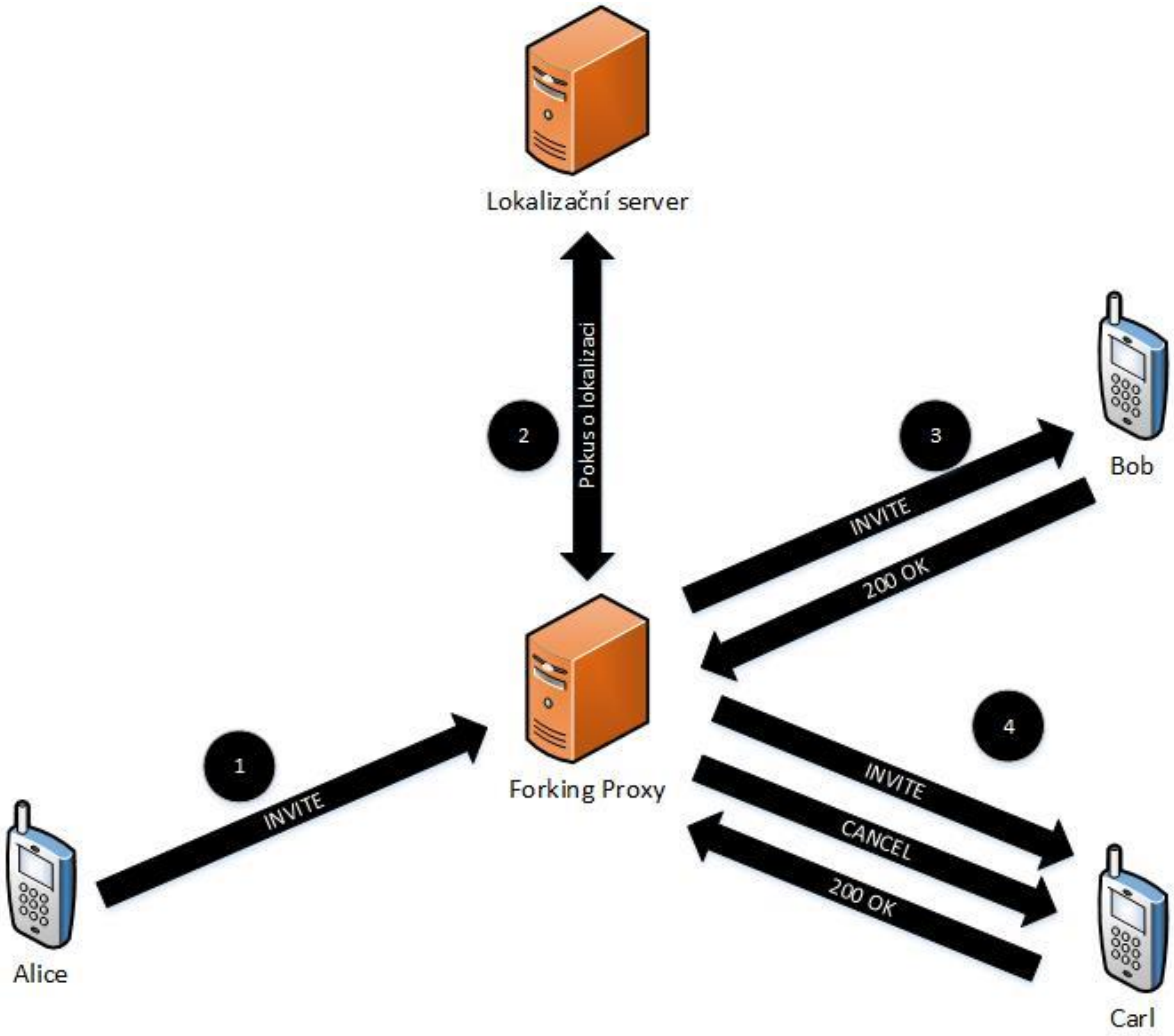
- B2BUA
- Proxy server
- TURN (*Traversal Using Relays around NAT*) server
- STUN (Session Traversal Utilities for NAT) server
- Udrženi otevřeného spojení pomocí keepalive

Atd.

Pokud bychom chtěli přidat úroveň zabezpečení pomocí IPSec, přibydou další obtíže, jelikož IPSec má ještě větší problémy s přechodem přes NAT (kvůli zaručení integrity dat nelze modifikovat IP adresy a porty jednotlivých protokolů) než SIP.

Ve zkratce – problém tkví v tom, že i v případě, že koncové zařízení chce udržovat otevřené spojení v paměti NAT, nemá žádný způsob, jak zjistit z běžného paketu, za jakou IP ho NAT v Internetu vydává. Jedno z řešení pro symetrický NAT je tedy to, že koncová stanice kontaktuje nějak server a ten jí sdělí, jaká je její veřejná IP. V případě překladu jedné veřejné IP za více vnitřních privátních IP a odlišení spojení na základě TCP/UDP portu, je však situace ještě komplikovanější.

Jak jsem řekl, je přechod přes NAT velké téma pro SIP, ale protože v mé diplomové práci jde především o IMS, není se třeba tím příliš zabývat. IMS totiž předpokládá, že všechny UE budou pracovat s IPv6 (RFC 6275 přidává do IPv6 další možnosti pro funkci v mobilních sítích). Pokud by však neexistovalo jiné řešení, může IMS síť obsahovat TURN nebo STUN server.



Obrázek 12 - Činnost Forking Proxy

4 Kamailio IMS

Kamailio je open source SIP server. Původně začal jako Open SIP Expres Router (OpenSER), ale časem byl přejmenován, aby se vyhnul problémům s názvem, který měly licencované jiné produkty [13].

Kamailio může fungovat jako SIP Proxy, Gateway, Redirect i Registrar server. Disponuje celou řadou rozšíření a zvládá až tisíce spojení hovorů. Na pluginech do Kamailia jsem i já postavil vlastní open-source IMS síť, resp. využil dostupný software a síť zprovoznil. Zní to jako poměrně snadný úkol, ale jak bude zřejmé z dalších kapitol, není to nic lehkého.

Kamailio pokračuje ve snahách jiného open source projektu a tím je Open IMS Core. Jedná se v podstatě o jedinou open source implementaci IMS. Zahrnuje P-CSCF, I-CSCF a S-CSCF a databázi HSS. Bohužel vývoj v podstatě ustal někde kolem roku 2013 (občas se sice nějaký zásah do kódu objeví i dnes, ale při mém testování mi taková změna rozbila HSS a musel jsem se vrátit ke starší verzi). U Kamailia vývoj probíhá i v oblasti IMS, ale díky velikosti celého projektu je mnohem těžší hledat problémy, které se vyskytnou při pokusu o zprovoznění.

Pro Kamailio neexistuje přímo určený HSS. Musel jsem tedy využít ten od Open IMS Core a poměrně bez problémů se mi podařilo komunikaci zprovoznit – což by nakonec nemělo být překvapení, vše by mělo dodržovat standardy.

V následujících kapitolách popíšu proces instalace Kamailio CSCF, HSS a také DNS serveru, protože bez něj se IMS síť neobejde.

Bohužel jsem v průběhu práce na IMS síti zjistil, že vývoj v open source komunitě moc neprobíhá a pokud jsou týmy, které se IMS zabývají, spíš vyvíjejí pro komerční sféru. Důvody se poměrně nabízí – IMS síť není malý projekt a vyvinout plně funkční síť na open source bázi vyžaduje mnoho času a lidí a to pochopitelně bez vidiny aspoň nějaké satisfakce v podobě peněz se nikomu příliš nechce. Navíc ani komunita, která by toto

využila, není tak rozsáhlá. Z diskuzí mi vyplynulo, že více či méně úspěšně se spíše starý Open IMS Core snaží zprovoznit pouze studenti doktorandského studia na různých univerzitách po světě a často i na jejich poměrně triviální dotazy nemá ani kdo odpovědět. Další lidé, kteří by open source IMS používali, jsou zaměstnanci telekomunikačních firem a operátorů. Jejich znalosti jsou pochopitelně dál než u studentů a nejspíše nedisponují ani takovým množstvím času, aby mohli po diskuzních fórech radit ostatním, co dělají špatně. Navíc mohou využívat testovacích platforem vlastních komerčních řešení, které dodavatel poskytne pro testování nových telefonů, služeb atd.

Samostatnou kapitolou je nedostatek open source klientů, přes které by se dala síť testovat. Těch existuje jen pár s různou kvalitou a vážně využít se dají v podstatě 3, když sečtu platformy Linux, Windows a Android. Vývoj na klientech dostupných na internetu skončil také někde kolem roku 2013.

4.1 Realizace sítě

Každý prvek sítě – tedy 3x CSCF, HSS a DNS server je nejlépe instalovat na vlastní operační systém, jinak řečeno virtuální OS. K tomu jsem využil open source software pro virtualizaci Oracle VM VirtualBox. Je třeba dát pozor na použité verze, protože starší verze 4 má problémy s Windows 10. Po upgradu na Windows 10 je tedy třeba použít novou verzi 5.

Všechny virtuální servery fungují na Linuxu. Volil jsem distribuce Ubuntu nebo Debian, ale nesnažil jsem se verze nijak unifikovat a používal jsem různé. Na funkci to však nemá vliv. Nejsem příliš zkušený v řešení problémů s Linuxem, a proto jsem volil takové distribuce, ke kterým lze dohledat na internetu co nejvíce informací. 5 virtuálních serverů byla poměrně zátěž pro můj PC, což se později nejspíše projevilo na některých charakteristikách při provedení hovoru. Na instalaci na 5 samostatných fyzických PC jsem však neměl ani čas ani zdroje. Standardně jsem strojům vyhradil 8 GB disk a 512 MB RAM. Při provozu virtuálních OS je dobré sledovat vytížení paměti RAM (přes programy „*top*“ nebo vylepšený „*htop*“) a případně tomu OS, který svoji paměť vyčerpal, přidat další.

Kromě vlastního software nutného pro IMS je ještě vhodné nainstalovat program Wireshark na analýzu síťového provozu, což je neocenitelná pomoc, buď při problémech, nebo při pozorování, jak fungují jednotlivé procesy. Druhou variantou je „tcpdump“.

Celá síť běží na privátních adresách. Program VirtualBox umožňuje udělat bridge (síťový most) všech síťových karet a také fyzické karty počítače a spojit je do jedné sítě. Potom spolu mohou komunikovat, ale důležitější je, že mají přístup i do internetu, což je téměř nutnost pro všechny instalace. Instalace bez internetového připojení by byl velmi zdlouhavý proces.

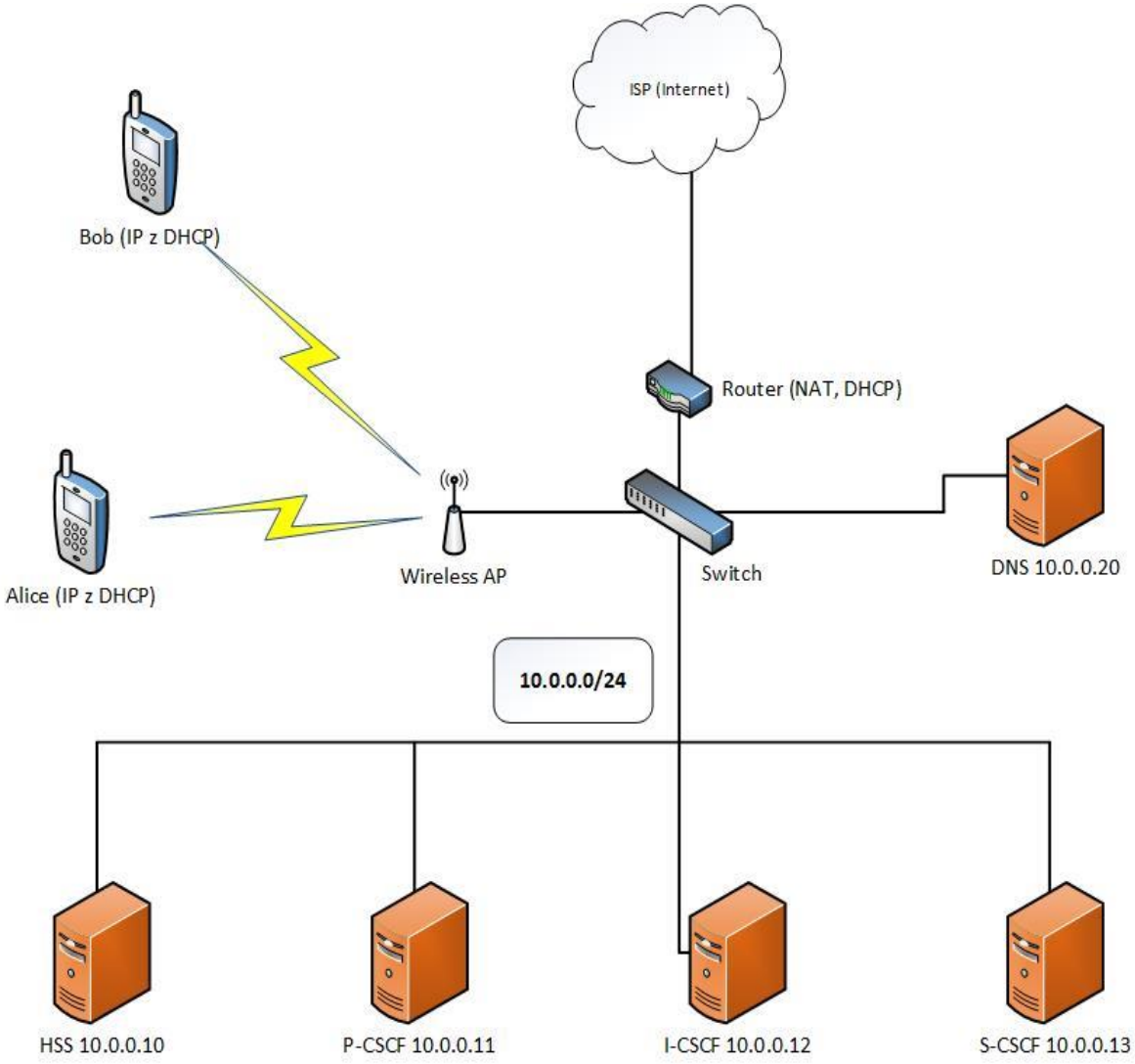
Pokud bychom chtěli, aby bylo možné provádět přes síť hovory i mimo LAN (*Local Area Network*), tak by musely být servery umístěny v internetu. V takovém případě by byla nejvhodnější varianta zvolit některého z poskytovatelů, kteří nabízí možnost nainstalovat virtuální stroje do jejich cloudového úložiště, jak to nabízí třeba Amazon pod Amazon Web Services (AWS). Vše samozřejmě za poplatek. Je možnost zde mít i potřebné DNS údaje.

Aby spolu virtuální servery mohly komunikovat, nejlehčím možným zapojením je umístit je všechny na jednu síť (subnet). Já jsem zvolil vlastní privátní síť, oddělenou od internetu díky NAT. Kromě dalších zařízení, které na síti jsou, je potřeba mít minimálně 4 volné IP adresy, ideálně však 7 pro otestování spojení. V mém případě byly adresy rozděleny podle tabulky 3, subnet 10.0.0.0/24. Pro ideální a trochu zajímavé testování se hodí použít 2 mobilní telefony či tablety s Android OS (Apple iOS jsem neměl příležitost otestovat) a aplikací IMS Droid – o klientech bude řeč v samostatných kapitolách. Další možností je poměrně dobrý (oproti jiným klientům) Boghe IMS Client pro Windows 7.

Zařízení	**IP adresa**
DNS server	10.0.0.20
HSS	10.0.0.10
P-CSCF	10.0.0.11
I-CSCF	10.0.0.12
S-CSCF	10.0.0.13
Smartphone s WIFI	Přidělená přes DHCP
Notebook s WIFI	Přidělená přes DHCP

Tabulka 3 - IP adresa

Na Obrázku 13 je logická realizace sítě. Vše by přesně takto mohlo fungovat, ale reálně veškeré servery jsou na jednom PC, AP a Switch jsou integrované do běžného WIFI routeru pro domácí použití.



Obrázek 13 - Logická topologie sítě

4.2 DNS server – bind9

Nutností pro fungování IMS sítě je DNS server, kde jsou uložené předklady z jmen jednotlivých entit na IP adresy. Vše by mohlo být uloženo na nějakém veřejném serveru, avšak je lepší mít ze vzdělávacího a praktického hlediska nad serverem kontrolu.

V případě, že by měla být tato pokusná síť spuštěna ve veřejném internetu, bylo by samozřejmě nutné mít i DNS překlady dostupné z internetu.

DNS server je v podstatě překladačem doménových jmen za IP adresy. V běžném životě překládá názvy stránek, které uživatel píše do adresního řádku prohlížeče. Nabízí však ze své podstaty několik funkcí, které se běžně užívají kvůli vedlejším benefitům. Prvním z nich je takzvaný load-balancing, kdy DNS server na DNS dotazy nevrací vždy jen jednu IP adresu, ale více různých, pod kterými se skrývají servery provádějící tu samou činnost, ale díky rozložení zátěže mohou zvládnout mnohem více požadavků, než by zvládnul jeden server [14]. Dalším využitím může být, pro IMS podstatné, odkázání na server vzhledem k jeho geografickému umístění – tedy aby nemuselo zařízení komunikovat se serverem až na druhé straně Země, ale nejlépe ve stejném státu. Toho IMS využívá pro přiřazení nejbližšího P-CSCF. Pro stejný efekt lze však využít distribuci P-CSCF po světě pomocí BGP (*Border Gateway Protocol* - protokol využitý pro směrování v Internetu), kterým dovoluje využít vždy nejbližší cílovou IP adresu uživateli. Dobře pozorovatelným efektem této schopnosti je například umístění DNS serveru Googlu 8.8.8.8, který pro uživatele v ČR je v České republice, ale obyvatele USA pod stejnou IP adresou do ČR nekomunikuje. Trochu to narušuje představu IP adres jako unikátního identifikátoru stanice internetu, pro BGP to však nepředstavuje žádný problém (dvě stejné cílové IP adresy se posoudí podle vzdálenosti autonomních systémů a že je možné dostat se do cíle více směry, není na závadu) [15].

Jako DNS server jsem využil nejpoužívanější BIND. Server umí mnoho věcí, pro IMS však stačí jen přidat soubor se zónou a v případě, že je potřeba i přístup do Internetu přes tento server, tak přidat ještě tzv. forwarder. Na ten se DNS server obrátí, pokud nebude znát odpověď sám, což bude v tomto případě téměř pořád. Stačí mu přidat DNS server, který byl využíván dosud, případně nějaký známý jako je 8.8.8.8.

Instalace začíná jako téměř u všeho Linuxového software na Debiantu nebo Ubuntu instalací pomocí balíčků. DNS server nainstalujeme příkazem na Obrázku 14.

```
apt-get install bind9
```

Obrázek 14 – Příkaz na instalaci BIND9 serveru

Aby server už fungoval jako cache pro internetové domény a mohly na něj odkazovat virtuální stroje, je potřeba mu nastavit, jakých IP adres se má ptát výše v hierarchii. To se nastaví v souboru /etc/bind/named.conf.options, kde je potřeba uvést aspoň jeden veřejný DNS server – Obrázek 15.

```
forwarders {
<IP adresa veřejného DNS serveru>;
};
```

Obrázek 15 – Nastavení veřejných DNS serverů

Jako další musíme do souboru „/etc/bind/named.conf.local“ přidat odkaz na umístění zónového souboru, kde jsou uloženy všechny informace o překladech jmen. Obsah souboru může vypadat tak jako na Obrázku 16.

```
zone “kamailio-ims.org” {
type master;
file “/etc/bind/kamailio-ims.org.dnszone”;
};
```

Obrázek 16 – Nastavení odkazu na DNS zónu

A nakonec je ještě nutné vytvořit výše zmíněný soubor se správným obsahem. Obsah souboru je na Obrázku 17 a soubor se tedy jmenuje „kamailio-ims.org.dnszone“.

Server se restartuje stejně jako velké množství jiných služeb na Linuxu (stejným způsobem se restartuje i samotné Kamailio) – Obrázek 18. Restart je nutné provést nejlépe vždy, když se změní některé nastavení.

Funkčnost lze ověřit buď klasickými příkazy jako je starší „*dnslookup*“, „*host*“ anebo novější „*dig*“, obvyklejším způsobem ale bude spíš „*ping*“ na jméno některé entity IMS – například „ping pcscf.kamailio-ims.org“, které by mělo adresu přeložit za správnou zvolenou. Pro funkci sítě je tento poslední krok naprosto zásadní a nelze bez něj pokračovat.

Troubleshooting problémů u DNS serveru pravděpodobně nenastane, ale v případě potíží by se daly kontrolovat běžící procesy, případně otevřenost portu 53 na serveru. Instalace a nastavení DNS serveru by však nemělo představovat žádný větší problém.

Jak je vidět z obrázku 17, IMS nepoužívá z DNS běžné A záznamy, ale NAPTR (*Name Authority Pointer*) a SRV (*Service Locator*).

```
$ORIGIN kamailio-ims.org.
$TTL 1W
@                       1D IN SOA       localhost. root.localhost. (
                                        2006101001      ; serial
                                        3H              ; refresh
                                        15M             ; retry
                                        1W              ; expiry
                                        1D )            ; minimum
                        1D      IN NS ns
ns                      1D      IN A 127.0.0.1

pcscf                   1D      IN A 10.0.0.11
_sip.pcscf              1D      SRV 0 0 5060 pcscf
_sip._udp.pcscf         1D      SRV 0 0 5060 pcscf
_sip._tcp.pcscf         1D      SRV 0 0 5060 pcscf

icscf                   1D      IN A 10.0.0.12
_sip                    1D      SRV 0 0 5060 icscf
_sip._udp               1D      SRV 0 0 5060 icscf
_sip._tcp               1D      SRV 0 0 5060 icscf

kamailio-ims.org.       1D IN A 127.0.0.1
kamailio-ims.org.       1D      IN NAPTR 10 50 "s" "SIP+D2U" "" _sip._udp
kamailio-ims.org.       1D      IN NAPTR 20 50 "s" "SIP+D2T" "" _sip._tcp

scscf                   1D      IN A 10.0.0.13
_sip.scscf              1D SRV 0 0 5060 scscf
_sip._udp.scscf         1D SRV 0 0 5060 scscf
_sip._tcp.scscf         1D SRV 0 0 5060 scscf

hss                     1D      IN A                                    10.0.0.10
```

Obrázek 17 – Nastavení samotné zóny

```
/etc/init.d/bind9 restart
```

Obrázek 18 – Restartování DNS serveru

4.3 Instalace P-CSCF, I-CSCF a S-CSCF

Ačkoliv existuje návod přímo na stránkách Kamailia, podle kterého by se zdálo, že instalace není žádný problém, sám jsem zjistil, že to tak vůbec není. Instalace těchto tří částí a řešení problémů nefunkčnosti mi totiž zabrala poměrně dost času.

První krok pro každou z tří částí spočívá v instalace Linuxové distribuce. Není sice řečeno která, ale doporučit lze Debian nebo Ubuntu. Jak již bylo řečeno, je vhodné využít virtuálních instalací, ale pokud je k dispozici dostatek fyzických serverů, bude výkon lepší.

Do každého serveru nejdříve musíme přidat odkaz na zdroje kódu, odkud se stáhne potřebný software. Několik kroků lze následovat z návodu na stránkách, kde nejdříve se přidá klíč, pak zdroje a nakonec se provede update dostupného software ze zdrojů:

```
wget -O - http://repository.ng-voice.com/PublicKey | apt-key add -

echo "deb http://repository.ng-voice.com wheezy ims rtpproxy" >> /etc/apt/sources.list

apt-get update
```

Dalším příkazem už můžeme nainstalovat Kamailio a všechny moduly:

```
apt-get install kamailio kamailio-ims-modules kamailio-presence-modules kamailio-tls-modules kamailio-xml-modules kamailio-xmlrpc-modules
```

Po tomto kroku jsou v adresáři „*/etc/kamailio/*“ konfigurační soubory a jednotlivé moduly v „*/usr/lib64/kamailio/modules*“. V dalším kroku je třeba stáhnout ze Githubu pro každý virtuální stroj, který bude představovat P-CSCF, I-CSCF a S-CSCF vlastní konfigurační soubor kamailio.cfg, pcscf.cfg, icscf.cfg nebo scscf.cfg a příslušný soubor.xml.

Soubory lze najít zde:

https://github.com/kamailio/kamailio/tree/master/examples

V tomto adresáři se nachází mnoho souborů, které poslouží jako návod nebo kontrola, některé části kódu jsou ale velmi dlouho neaktualizované. Ze složky /etc/kamailio/ je tedy potřeba smazat soubor kamailio.cfg a nahradit do souborem se stejným názvem, ale ze správné složky podle toho, který prvek sítě nastavujeme. V samotném souboru (pcscf.cfg/icscf.cfg/ scscf.cfg) je potřeba udělat několik věcí.

Je třeba změnit IP adresy a porty, na kterých bude server naslouchat síťovému provozu. Pro základní funkci je třeba mít funkční především TCP a UDP porty, ale není důvod nenastavit například i TLS. Dále je třeba nastavit alias, tzn. doménové jméno entity a různé identifikátory NETWORKNAME, HOSTNAME atp. Následuje několik položek, které zapínají určité další funkce, zapisují se jako #!define <název>. Zakomentovat položku lze přidáním druhého křížku na začátek. V souboru kamailo.cfg není potřeba měnit nic, ale jeho obsah může posloužit pro případné hledání problémů, jelikož SIP odpovědi, které lze vidět ve Wiresharku, lze přímo dohledat zde a najít odtud další části kódu.

Další důležitou složkou na Githubu je kamailio/modules. Zde je zdrojový kód všech modulů a dokonce se na nich i občas pracuje a jsou nové změny a opravy. Nejdůležitější je soubor README u každého modulu. Jsou zde popsány jednotlivé parametry a moduly. Bohužel jak je z dat aktualizací zřejmé, soubor README byl aktualizovaný v některých případech před 3 lety, ale práce na ostatních souborech pokračovala dál. Proto se mi pak stalo, že v README bylo napsáno, že standardní hodnota pro určitou funkci je „vypnuto“, ale v instalaci byla tato funkce zapnutá a pokusy o spuštění selhávaly. Z této složky však není třeba nic stahovat.

Spuštění Kamailia se provede pomocí:

```
/etc/init.d/kamailio restart
```

To však zahlásí chybu a musí se nejdříve upravit soubor /etc/default/kamailio. Tím začne poměrně dlouhý koloběh různých chyb a problémů, ačkoliv návod na internetu vypadá na snadnou instalaci. Další pokus o spuštění ohlásí chybu s databází, resp. že Kamailio nemohlo nalézt modul db_mysql. Ten v modulech opravdu není, P-CSCF a S-CSCF ho ani nepotřebují, takže stačí v pcscf.cfg a scscf.cfg zakomentovat pomocí # řádek #!define DB_URL. Pro I-CSCF, které databázi potřebuje, je třeba nainstalovat mysql server a poté spustit nainstalovalý balík „kamailio-mysql-modules“. Navíc je na serveru, kde běží I-CSCF, ještě třeba udělat „mysql -uroot -p < icscf.sql“, což vytvoří databázi. Soubory .xml slouží pro komunikaci s HSS a jejich nastavení popíšu v části s instalací HSS.

Dále je třeba vytvořit soubor dispatcher.list, který je vyžadován některými moduly pro load-balancing a anti-flooding. Stačí ho tedy vytvořit. Potřebu vytvoření souboru můžeme sledovat v chybových hláškách objevujících se v „*/var/log/syslog*“. Po tomto kroku už by Kamailio mělo běžet, což můžeme ověřit běžícími procesy pomocí programu „top“. Velice podobný proces tedy funguje pro všechny 3 části.

```
touch /etc/kamailio/dispatcher.list
```

4.4 Instalace HSS

HSS není dostupný společně s CSCF Kamailia a je třeba použít jiný. Já se pokusil využít a propojit Kamailio prvky a HSS od Open IMS Core. Pro instalaci HSS je dostupná lepší dokumentace než pro Kamailio a lze ho tedy použít bez problémů i s dostupným návodem. Oproti návodu není třeba instalovat vše, ale jen samotný HSS.

Před samotnou prací je třeba nainstalovat do Linuxové distribuce nutný software. Jedná se o:

- GCC,
- JDK 1.5 a vyšší,
- MySQL,
- Bison, flex,
- libxml2-dev a libmysql++-dev,
- Curl a libcurl,
- Subversion.

Pro HSS vytvoříme složku v /opt/ a přepneme se do ní.

Lze samozřejmě rovnou vytvořit složku pouze pro HSS, ale při první instalaci je vhodné se zdržet vlastních modifikací, které přinesou spíše problémy.

```
mkdir /opt/OpenIMSCore

cd /opt/OpenIMSCore
```

Při instalaci kompletního OpenIMSCore by se instalovaly do složky *ser_ims* všechny CSCF části, to zde však přeskočíme a vytvoříme rovnou složku pro HSS a do ní stáhneme zdrojový kód.

```
mkdir FHoSS

svn checkout https://svn.code.sf.net/p/openimscore/code/ser_ims/trunk ser_ims
```

Výhodou toho, že autoři využili program Subversion je, že pokud některá verze nefunguje a je podezření na bug, který se objevil s novější verzí, lze jednoduše zkusit starší verzi (revision). Jak je příkaz uvedený výše, stáhne se automaticky ta nejnovější, tedy 1195. S tou jsem měl problém, který jsem dohledal až zpět k revision 1191, kde proběhly úpravy

přesně té části kódu, se kterou jsem měl problémy. Proto jsem využil revision 1190. Starší revision se stáhne pomocí přepínače *–r <revision>* tedy:

```
svn checkout –r 1190 https://svn.code.sf.net/p/openimscore/code/ser_ims/trunk ser_ims
```

Tím máme stažený zdrojový kód v adresáři FHoSS.

Největší potíže při instalaci HSS jsou s Javou. Verze musí být novější než 1.5. V dostupném návodu je navíc starší Sun Java, kterou bylo obtížné najít. Využít se dá ale i novější verze. Já používal tuto verzi:

```
root@HSS# java -version

java version "1.7.0_80"

Java(TM) SE Runtime Environment (build 1.7.0_80-b15)

Java HotSpot(TM) 64-Bit Server VM (build 24.80-b11, mixed mode)
```

HSS se zkompiluje v adresáři FHOSS následujícími příkazy:

```
cd FHoSS

ant compile

ant deploy

cd ..
```

Po instalaci je třeba ještě zeditovat několik souborů podle vlastních preferencí a plánu na provoz HSS. Prvním z nich je v adresáři /opt/FHoSS/deploy/ soubor DiameterPeerHSS.xml.

V něm je potřeba změnit položku *FQDN* (*Fully Qualified Domain Name*) podle vlastní domény (pokud chceme použít standardní doménu, mohou všechny *FQDN* a další navazující názvy zůstat. Stejné podmínky platí pro položku *Realm*. Další je třeba nastavit

dva řádky <Peer FQDN...>, kde se také zapíše FQDN a Realm, ale tentokrát pro I-CSCF a S-CSCF. A je potřeba nezapomenout port, na kterém obě entity naslouchají (ten se nastavuje v Kamailiu v souborech icscf.xml a scscf.xml). Jako poslední je třeba nastavit port a IP, na kterém bude poslouchat HSS pod položkou *Acceptor*. Tím jsou připravené síťové funkce HSS.

Dále v souboru hss.properties můžeme změnit, kde bude naslouchat webové rozhraní, kterým HSS disponuje. Standardně je nastavená IP na 127.0.0.1 a port 8080. Toto je třeba změnit, pokud bychom chtěli administrovat HSS z jiného PC než je ten lokální, tedy z LAN nebo Internetu. Webové rozhraní HSS běží od chvíle, kdy se HSS úspěšně spustí.

Než se tak může stát, je velmi pravděpodobné, že budeme mít problém s Javou a jejím umístěním. Hlavním problémem je, že ve spouštěcím skriptu startup.sh je uvedená jiná cesta, než kterou budeme mít my, konkrétně jde o systémovou proměnnou JAVA_HOME. Tu v Linuxu zjistíme pomocí:

```
echo $JAVA_HOME
```

Pokud se vrátí prázdný řádek, není nastavená vůbec. Cestu nastavíme takto:

```
export JAVA_HOME=<cesta>
```

Poté můžeme znova spustit příkaz *echo* a měli bychom dostat nějaký výstup. Problém s tímto manuálním nastavením je, že se musí udělat s každým spuštěním HSS. Proto je lepší ho rovnou zahrnout do skriptu *startup.sh*. V mém případě pak tedy modifikace skriptu vypadala takto:

```
export JAVA_HOME=/usr/lib/jvm/java-7-oracle/jre

$JAVA_HOME/bin/java –cp $CLASSPATH de.fhg.fokus.hss.main.HSSContainer $1 $2 $3 $4 $5 $6 $7 $8 $9
```

Konkrétní cesta samozřejmě záleží na verzi a umístění Javy.

V dalším kroku už můžeme HSS spustit příkazem:

```
./startup.sh
```

Terminál, ve kterém HSS běží, poté vypisuje informace s různými hláškami, užitečnými pro debugging. Pokud nemáme nastartované I-CSCF a S-CSCF, tak minimálně se budou periodicky objevovat chybové zprávy se selháním připojení. Tyto zprávy hned můžeme použít pro hledání problémů, jelikož vypisují konkrétní porty atd.

Teď už lze také spustit webové rozhraní HSS. V něm lze přidávat, mazat a modifikovat jednotlivé uživatele. Několik změn je potřeba také udělat, pokud se rozhodneme změnit původní doménu za vlastní. Do webového rozhraní se dostaneme přes adresu, kterou jsme mohli modifikovat hss.properties. Pokud se tak nestalo, stačí do prohlížeče zadat 127.0.0.1:8080 a v následujícím formuláři zadat jméno „hssAdmin“ a heslo „hss“. Po zadání hesla uvidíme webové prostřední jako je na obrázku 19.

Obrázek 19 - Webové rozhraní HSS

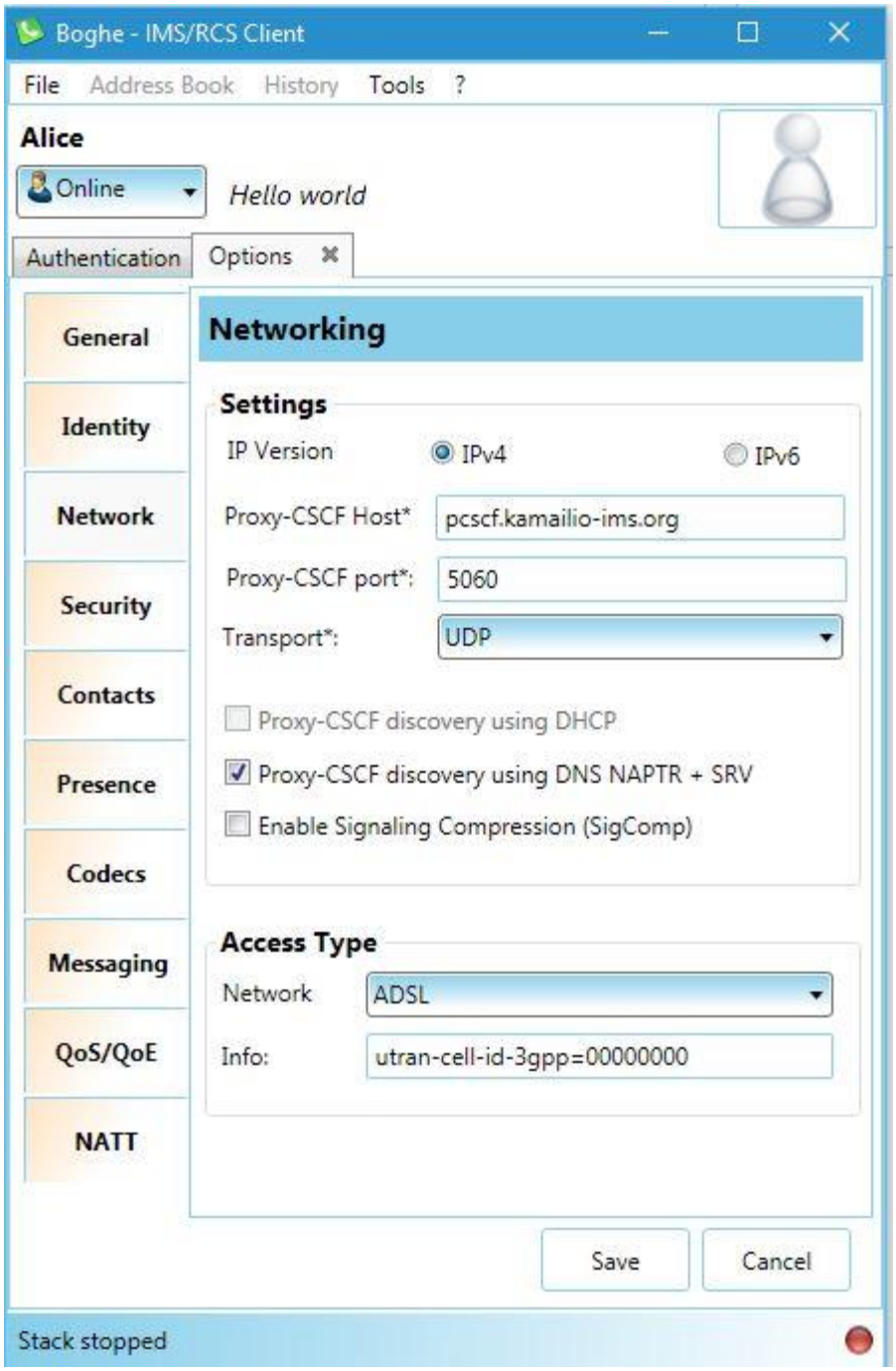
4.5 IMS klienti

4.5.1 Boghe IMS client

Jedná se o klient pro Windows 7/8 od Doubango Telecom [15]. Existuje také mobilní verze. Sám jsem zkoušel klienta na Windows 7 a 10, kde kromě občasných pádů funguje celkem dobře. Má poměrné široké možnosti nastavení a kromě toho je i celkem uživatelsky přívětivý. Na Obrázku 20 je přihlašovací obrazovka klienta už s vyplněnými údaji pro moji síť. Při přihlášení probíhají procesy tak, jak jsou popsané pro registraci atd. – vše lze sledovat ve wiresharku. Pro příklad nastavení je na Obrázku 21 nastavení síťových údajů. Zde lze nastavit zjištění P-CSCF buď přes DNS nebo lze zadat přímo IP adresu. Lze změnit také port – P-CSCF může v síti naslouchat i na jiných portech (Open IMS Core například používá UDP porty pro P-CSCF 4060, pro I-CSCF 5060 a S-CSCF 6060). Popis přenastavení je uveden v předchozích kapitolách, v souboru *kamailio.cfg*. Je potřeba dále nastavit jméno uživatele, public a private (což je běžně public identity ale bez „sip:“) identity, heslo a realm (tedy síť).

Bohužel projekt v podstatě od prosince 2013 nepokračuje. I když byl nedávno s ukončením Google Code přesunut na Github, žádnou velkou aktivitu vývojářů jsem nezaznamenal – většina souborů má poslední aktualizaci před 3 měsíci, kterou způsobilo přesunutí na Github.

Obrázek 20 – Přihlášení do Boghe klienta

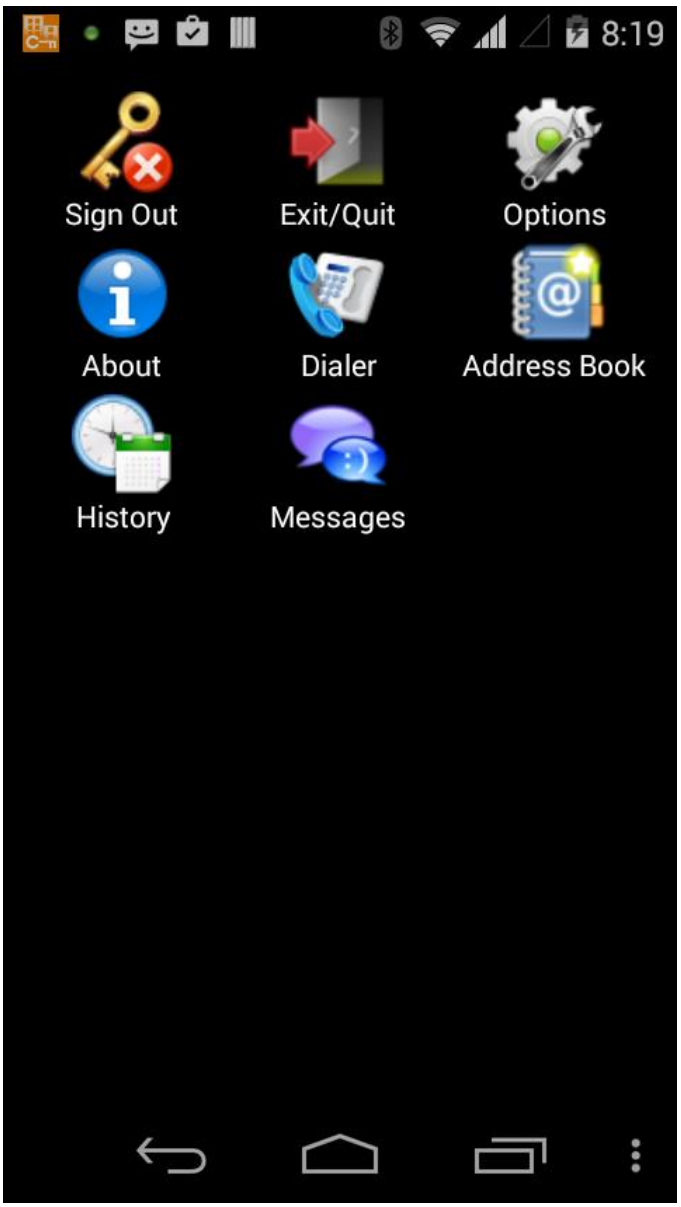


Obrázek 21 – Nastavení sítě pro Boghe IMS klienta

4.5.2 IMS Droid

Stejně jako předchozí klient je i tento od Doubango Telecom. Na rozdíl od Boghe je ale IMS Droid určený pro operační systém mobilních telefonů Android. Funguje snad ze všech

vyzkoušených klientů napříč platformami nejlépe. Lze díky němu simulovat i telefonní síť, pokud se využijí dva telefony na WIFI síti. Včetně audia dokáže přenášet i video. Bohužel stejně jako Boghe i jeho poslední verze je z prosince 2013 a nezdá se, že by kromě přesunutí projektu na Github existovala nějaká další činnost vývojářů. Nastavení probíhá v podstatě stejně jako Boghe, pouze v jiném uživatelském prostředí Androidu. Na Obrázku 22 je hlavní menu.



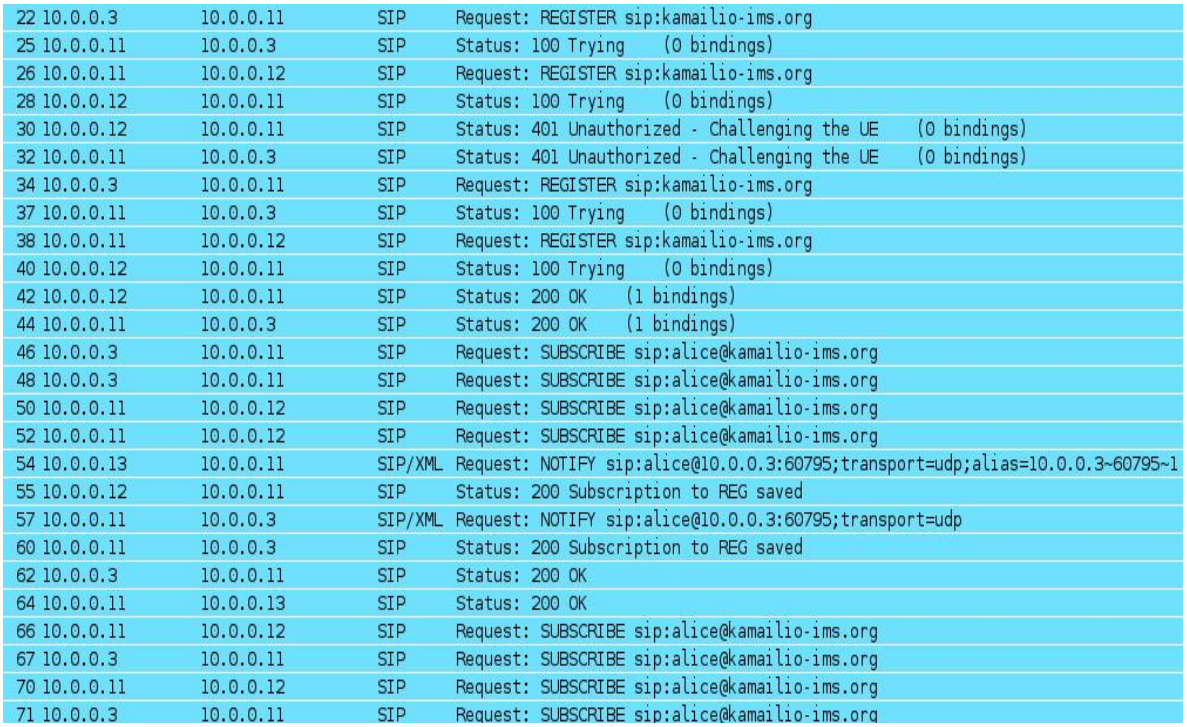
Obrázek 22 – Hlavní menu IMS Droid

4.5.3 myMonster

MyMonster klienta jsem používal na Linuxu. Už nepochází od Doubango Telecom a na instalaci a provozu je to znát. Instalace je zdánlivě snadná, stačí stáhnout zdrojové soubory a spustit. To však na běžné distribuci nezafunguje a myMonster vyžaduje velké množství dalších knihoven a závislostí, některé z nich už jsou zastaralé. Nakonec se klienta podaří rozběhnout, ale ani pak nefunguje bezchybně. Pokud z jakéhokoliv důvodu selže registrace, klient stále vypadá, že se snaží přihlásit. Nejlepší řešení je ukončit celý program (například Ctrl+c z terminálu) a spustit ho znovu. Používal jsem ho především na otestování registrace, pro samotné telefonování jsem využíval Boghe klienta z notebooku a IMS Droid přes mobilní telefon.

4.6 Registrace a hovor

Registrace klienta, resp. uživatele do sítě probíhá celkem tak, jak by se dalo očekávat podle výše popsaných postupů v kapitolách o IMS procesech a SIP. Uvedu tedy pouze obrázek, jak registrace může vypadat v programu Wireshark na P-CSCF. IP adresy jsem uvedl v kapitole 4.1, takže jen pro doplnění – IP adresa 10.0.0.3 je můj systém Windows 10, ve kterém jsem spustil Boghe IMS Client a registroval se do IMS běžících ve virtuálních systémech na tom stejném počítači.



22	10.0.0.3	10.0.0.11	SIP	Request: REGISTER sip:kamailio-ims.org
25	10.0.0.11	10.0.0.3	SIP	Status: 100 Trying (0 bindings)
26	10.0.0.11	10.0.0.12	SIP	Request: REGISTER sip:kamailio-ims.org
28	10.0.0.12	10.0.0.11	SIP	Status: 100 Trying (0 bindings)
30	10.0.0.12	10.0.0.11	SIP	Status: 401 Unauthorized - Challenging the UE (0 bindings)
32	10.0.0.11	10.0.0.3	SIP	Status: 401 Unauthorized - Challenging the UE (0 bindings)
34	10.0.0.3	10.0.0.11	SIP	Request: REGISTER sip:kamailio-ims.org
37	10.0.0.11	10.0.0.3	SIP	Status: 100 Trying (0 bindings)
38	10.0.0.11	10.0.0.12	SIP	Request: REGISTER sip:kamailio-ims.org
40	10.0.0.12	10.0.0.11	SIP	Status: 100 Trying (0 bindings)
42	10.0.0.12	10.0.0.11	SIP	Status: 200 OK (1 bindings)
44	10.0.0.11	10.0.0.3	SIP	Status: 200 OK (1 bindings)
46	10.0.0.3	10.0.0.11	SIP	Request: SUBSCRIBE sip:alice@kamailio-ims.org
48	10.0.0.3	10.0.0.11	SIP	Request: SUBSCRIBE sip:alice@kamailio-ims.org
50	10.0.0.11	10.0.0.12	SIP	Request: SUBSCRIBE sip:alice@kamailio-ims.org
52	10.0.0.11	10.0.0.12	SIP	Request: SUBSCRIBE sip:alice@kamailio-ims.org
54	10.0.0.13	10.0.0.11	SIP/XML	Request: NOTIFY sip:alice@10.0.0.3:60795;transport=udp;alias=10.0.0.3~60795~1
55	10.0.0.12	10.0.0.11	SIP	Status: 200 Subscription to REG saved
57	10.0.0.11	10.0.0.3	SIP/XML	Request: NOTIFY sip:alice@10.0.0.3:60795;transport=udp
60	10.0.0.11	10.0.0.3	SIP	Status: 200 Subscription to REG saved
62	10.0.0.3	10.0.0.11	SIP	Status: 200 OK
64	10.0.0.11	10.0.0.13	SIP	Status: 200 OK
66	10.0.0.11	10.0.0.12	SIP	Request: SUBSCRIBE sip:alice@kamailio-ims.org
67	10.0.0.3	10.0.0.11	SIP	Request: SUBSCRIBE sip:alice@kamailio-ims.org
70	10.0.0.11	10.0.0.12	SIP	Request: SUBSCRIBE sip:alice@kamailio-ims.org
71	10.0.0.3	10.0.0.11	SIP	Request: SUBSCRIBE sip:alice@kamailio-ims.org

Obrázek 23 – Průběh registrace SIP zpráv na P-CSCF

Hovor oproti registraci je ještě o něco málo ilustrativní, proto obrázek z Wiresharku nebudu uvádět. Je navíc popsán v podstatě přesně u sestavení hovoru v předchozích kapitolách.

5 Závěr

IMS síť založenou na Call Session Control Function z Kamailio a HSS z původního OpenIMSCore jsem nainstaloval a spustil úspěšně. Přes síť lze telefonovat různými klienty, jdou registrovat uživatelé, lze další přidávat a upravovat. Síť umožňuje kromě přenosu audia i přenos videa. Do tohoto momentu jsem tedy větší část zadání splnil. Problém se však ukázal v dalších prvcích, které by síť mohla obsahovat a o nichž se mluví i v zadání. Oproti původním očekáváním se ukázalo, že ačkoliv třeba Kamailio je připraveno na několik dalších prvků sítě, tak samotný software není dostupný v open source podobě. Vlastní doprogramování by byla práce na mnohem delší dobu, než je samotný rozsah diplomové práce. Navíc já osobně nejsem nijak zdatný programátor a žádné velké programátorské schopnosti ani původně při domlouvání zadání neměly být potřeba. Programování například BGCF nebo PCRF by bylo vhodnou náplní doktorandského studia, na němž by bylo podle mého názoru dostatečně času na takovou náročnou aktivitu. Osobně vidím největší problém v tom, že vybudovat IMS síť ve volném čase je velmi náročné a lidí, kteří by měli dostatečné schopnosti, je malé množství. Proto se jejich snahy nejspíše soustředí na vývoj vlastních komerčních řešení, protože trávit desítky hodin prací zadarmo na open source si podle mého názoru nemohou dovolit.

Pokud by se takováto open source síť podařila doplnit dalšími prvky, mohlo by být velmi zajímavé vyzkoušet ji v reálném prostředí telefonní sítě. To by však vyžadovalo spolupráci s aspoň jedním operátorem a spuštění v jeho síti (i když určitě jen v testovacích podmínkách). Jako nahrazení reálné sítě se dá dost dobře využít i Wifi.

Vlastní hovory trpěly poněkud větším zpožděním hlasu. Nejsem schopen přesně říct, kde se toto zpoždění vytvářelo, ale jako první bych hledal problém ve faktu, že síť běží na 5 virtuálních strojích, které jsou všechny spuštěny ve VirtualBoxu. Myslím si, že by velké zlepšení nastalo, pokud by se síť přesunula na reálné servery nebo aspoň středně výkonné počítače připojené do jedné sítě. Další alternativou je spouštět virtuální počítače na software, který je k tomu určený, tedy třeba je vytvořit pomocí hypervisoru vSphere od

VMWare nebo Hyper-V od Microsoftu. Další příčinou zpoždění může být také nevyladěnost klientů nebo i serverové části (CSCF a HSS). Klienti bohužel nejsou zdaleka dokončeni a většina z nich by se dala spíše považovat za betaverze, než za hotový program. Pády a různá zaseknutí programu jsou na denním pořádku.

Sám o sobě je koncept IMS velmi zajímavý a osobně doufám, že se k němu naše telekomunikační sítě postupně dostanou. I když už dnes lze běžně komunikovat přes tzv. „mobilní data“ mezi mobilním telefonem a počítačem, tak až přesun všech zařízení na IPv6 a využití jednotné platformy a protokolů pro komunikace umožní, že nebude naprosto záležet, jakým způsobem se člověk připojí a bude moct volně komunikovat s kýmkoliv jiným. V dnešní době stále není běžné, že by člověk od počítače snadno telefonoval s někým, kdo jde po ulici a používá běžný hlasový tarif operátora. Možné to samozřejmě je už i dnes, ale jde často o zbytečně složitá řešení, která se dodávají běžně do call-center a firem.

Použité zkratky

3G	Síť třetí generace
3GPP	3rd Generation Partnership Project
4G	Síť čtvrté generace
ALG	Application Level Gateway
AP	Access Point
API	Application Programming Interface
AS	Application Server
AuC	Authentication Center
B2BUA	Back-to-back User Agent
BGCF	Breakout Gateway Control Function
BGP	Border Gaterway Protocol
BRAS	Broadband Remote Access Server
CS	Circuit Switched
CSCF	Call Session Control Function
DNS	Domain Name System
DSL	Digital Subscriber Line
DSLAM	Digital Subscriber Line Access Multiplexer
FQDN	Fully Qualified Domain Name
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
GRUU	Globally Routable User Agent URI
GSM	Global Systém for Communication
HLR	Home Location Register
HSS	Home Subscriber Server
HTTP	Hypertext Transfer Protocol
I-CSCF	Interrogating-Call Session Control Function
IM	Instant Messaging
IMS	IP Multimedia Subsystem
IMS-MGW	IMS Media Gateway
IP	Internet Protocol
IPv6	Internet Protocol version 6
IPSec	IP Security
ISDN	Integrated Services Digital Network
ISIM	IMS Identity Module

ISUP	ISDN User Part
ITU-T	International Telecommunication Union-Telecommunication
LAN	Local Area Network
LTE	Long Term Evolution
MGCF	Media Gateway Control Function
MPLS	Multiprotocol Label Switching
MRF	Media Resource Function
MRFC	Media Resource Function Controller
MRFP	Media Resource Function Procesor
MGW	Media Gateway
MTU	Maximum Transmission Unit
NAI	Network Access Identifier
NAPTR	Name Authority Pointer
NAT	Network Address Translation
OS	Operation System
OSI	Open System Interconnection
P-CSCF	Proxy-Call Session Control Function
PCRF	Policy and Charging Rules Function
PPPoE	Point-to-Point Protocol over Ethernet
PS	Packet Switched
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RADIUS	Remote Authentication Dial-In User Service
RAN	Radio Access Network
RTP	Real-time Transport Protocol
S-CSCF	Session-Call Session Control Function
SA	Security Associations
SAR	Server-Assignment-Request
SDP	Session Description Protocol
SGSN	Serving GPRS Support Node
SGW	Signalling Gateway
SigComp	Signalling Compression
SIP	Session Initiation Protocol
SIP URI	Session Initiation Protocol Uniform Resource Identifier
SLF	Subscriber Location Function
SRTP	Secure Real-time Transport Protocol
SRV	Service Record
STUN	Session Traversal Utilities for NAT

TCP	Transmission Control Protocol
TLS	Transport Layer Security
TrGW	Transition Gateway
TTL	Time-to-live
TURN	Traversal Using Relays around NAT
UAR	User-Authorization-Request
UDP	User Datagram Protocol
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
VoLTE	Voice over Long Term Evolution
VPN	Virtual Private Network
WIFI	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
xDSL	(SH/A/V) Digital Subscriber Line
ZZS	Zdravotnická Záchranná Služba

Seznam obrázků

Seznam tabulek

Použitá literatura

[1] Singtel, Samsung and Ericsson unveil world's first full-featured Voice over LTE service. Singtel [online]. Singapore, 2014 [cit. 2016-01-02]. Dostupné z: http://www1.singtel.com/about-us/news-releases/singtel-samsung-and-ericsson-unveil-worlds-first-full-featured-voice-over-lte.html

[2] T-Mobile. T-MOBILE KOMERČNĚ SPUSTIL VOLTE JAKO PRVNÍ OPERÁTOR V ČR. T-Mobile Tiskove Centrum [online]. Praha, 2015 [cit. 2016-01-02]. Dostupné z: http://t-press.cz/cs/tiskove-materialy/tiskove-zpravy-t-mobile/t-mobile-komercne-spustil-volte-jako-prvni-operator-v-cr.html

[3] Releases Description. Release 14 [online]. [cit. 2016-01-02]. Dostupné z: http://www.3gpp.org/ftp/Information/WORK_PLAN/Description_Releases/

[4] POIKSELKÄ, Miikka a Georg MAYER. The IMS: IP multimedia concepts and services. 3rd ed. Chichester, U.K.: Wiley, 2009, xxviii, 502 p. ISBN 0470721960.

[5] CAMARILLO, Gonzalo a Miguel A GARCÍA-MARTÍN. The 3G IP multimedia subsystem (IMS): merging the Internet and the cellular worlds. 2nd ed. Hoboken, NJ: J. Wiley & Sons, 2006, xxvi, 427 s. ISBN 0-470-01818-6.

[6] AHSON, Syed a Mohammad ILYAS. IP multimedia subsystem (IMS) handbook. Boca Raton: CRC Press, 2009, xv, 543 p. ISBN 1420064592.

[7] The Network Access Identifier [online]. 2005 [cit. 2016-01-02]. Dostupné z: https://tools.ietf.org/html/rfc4282

[8] SIP: Session Initiation Protocol. RFC 3261 [online]. 2002 [cit. 2016-01-02]. Dostupné z: https://www.ietf.org/rfc/rfc3261.txt

[9] SDP: Session Description Protocol. RFC 4566 [online]. [cit. 2016-01-02]. Dostupné z: https://tools.ietf.org/html/rfc4566

[10] JOHNSTON, Alan B. SIP: understanding the Session Initiation Protocol. 2nd ed. Boston: Artech House, 2004, xxiii, 283 p. Artech House telecommunications library. ISBN 1-58053-655-7.

[11] STALLINGS, William. The Session Initiation Protocol [online]. 2003 [cit.2016-01-02]. Dostupné z URL: http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_6-1/sip.html.

[12] Time to Live (TTL). Requirements for IP Version 4 Routers [online]. 1995 [cit. 2016-01-02]. Dostupné z: https://tools.ietf.org/html/rfc1812#page-85

[13] SIP Express Router. Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2015 [cit. 2016-01-02]. Dostupné z: https://en.wikipedia.org/wiki/SIP_Express_Router

[14] A Border Gateway Protocol 4 (BGP-4). RFC 4271 [online]. 2006 [cit. 2016-01-02]. Dostupné z: https://www.ietf.org/rfc/rfc4271.txt

[15] DNS Support for Load Balancing. RFC 1794 [online]. [cit. 2016-01-02]. Dostupné z: https://tools.ietf.org/html/rfc1794

[16] Boghe Client Download. Google Code Archive [online]. 2013 [cit. 2016-01-02]. Dostupné z: https://code.google.com/archive/p/boghe/downloads