

Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky

Rozšírenia SIP protokolu pre účely
vzdialenej asistencie v mobilných sieťach

Diplomová práca

2013

Bc. Marek Šuščák

Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky

**Rozšírenia SIP protokolu pre účely
vzdialenej asistencie v mobilných sieťach**

Diplomová práca

Študijný program: Informatika
Študijný odbor: Informatika
Školiace pracovisko: Katedra počítačov a informatiky (KPI)
Školiteľ: doc. Ing. Jaroslav Porubän, PhD.
Konzultant: doc. Ing. Jaroslav Porubän, PhD.

Košice 2013

Bc. Marek Šuščák

Abstrakt v SJ

Táto diplomová práca sa zaoberá problematikou rozšírenia protokolu SIP a súvisiacich protokolov pre účely zefektívnenia vzdialenej asistencie osôb v mobilných sieťach. Nadväzuje pritom na nedávny výskum v oblasti vzdialenej asistencie zrakovo postihnutých osôb, ktorý bol uskutočnený na Katedre počítačov a informatiky, FEI TU v Košiciach v rámci projektu Mapz. Za týmto účelom sa konkrétne zameriava na nájdenie štandardizovaného spôsobu obohatenia protokolov SDP a RTP o možnosti posielania prídavných informácií obsiahnutých priamo v multimediálnom toku obrazových alebo zvukových údajov. V úvodnej časti práca stručne vyhodnocuje aktuálny stav výskumu v projekte Mapz, z ktorého vyplývajú návrhy na zlepšenie uvedené v tejto práci, definuje základné pojmy a bližšie sa venuje teoretickým poznatkom o štruktúre a princípe fungovania protokolu SIP v spolupráci so súvisiacimi protokolmi v zmysle služieb VoIP potrebných pre vzdialenú asistenciu v mobilných sieťach. Analytická časť poukazuje na možnosti rozšírenia jednotlivých protokolov a analyzuje existujúce implementácie robustného protokolu SIP pre mobilné zariadenia. V praktickej časti práca opisuje vývoj rozšírení protokolov SDP a RTP určených pre existujúci aplikačný rámec Doubango. Vyvíjané rozšírenia zakladá na navrhovanom štandarde RFC 5285 publikovanom členmi komunity IETF v júli 2008. Funkčnosť navrhnutého riešenia demonštruje na vyvinutej mobilnej aplikácii. Záver práce je venovaný experimentálnym testom miery zdokonalenia postupov oproti súčasnemu stavu.

Kľúčové slová

Vzdialená asistencia, zrakovo postihnutý, nevidiaci, protokol, RFC 5285, SIP, SDP, RTP, rozšírenia hlavičiek.

Abstrakt v AJ

This master's thesis deals with the issues of extending SIP and related protocols for the purpose of streamlining the remote assistance of people in mobile networks. It builds on top of recent research made in field of remote assistance of visually impaired people, which was carried out at the Department of Computers and Informatics, FEI TU of Košice as a part of project Mapz. For that purpose, it specifically focuses on finding a standardized way to enhance SDP and RTP protocols to provide possibility of sending additional information included directly in multimedia stream of video or audio data. In the introductory section, the work briefly describes current state of the research made as a part of Mapz project which resulted in subsequent improvements discussed within this work, defines the basic terms and further deals with theoretical knowledge on the structure and principle of operation of SIP in collaboration with related protocols in terms of VoIP services needed for remote assistance in mobile networks. The analytical part points out the opportunities of extending individual protocols and analyzes the existing implementations of robust SIP protocol for mobile devices. In the practical part it describes the development of SDP and RTP protocols extensions for the existing Doubango framework. The extensions being developed are based on proposed standard RFC 5285 published by the members of IETF in July 2008. Functionality of proposed solution is demonstrated on the developed mobile application. The conclusion of this work is dedicated to measuring the rate of improvement over the current research state based on the practical experiment.

Klíčové slová v AJ

Teleassistance, remote assistance, remote guidance, visually impaired, blind, protocol, RFC 5285, SIP, SDP, RTP, header extensions.

Zadanie práce

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
Katedra počítačov a informatiky

ZADANIE DIPLOMOVEJ PRÁCE

Študijný odbor: **9.2.1 Informatika**
Študijný program: **Informatika**

Názov práce:

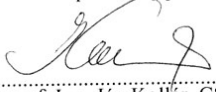
Rozšírenia SIP protokolu pre účely vzdialenej asistencie v mobilných sieťach
SIP Protocol Extensions for Teleassistance in Mobile Networks

Študent (tituly, meno, priezvisko): **Bc. Marek Šuščák**
Školiteľ (tituly, meno, priezvisko): **doc. Ing. Jaroslav Porubän, PhD.**
Školiace pracovisko: **Katedra počítačov a informatiky**
Konzultant práce (tituly, meno, priezvisko):
Pracovisko konzultanta:

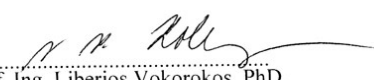
Pokyny na vypracovanie diplomovej práce:

1. Analyzovať a vyhodnotiť možnosti rozšírenia existujúcich riešení využívajúcich služby SIP protokolu.
2. Navrhnuť metódu využívajúcu rozšírenia SIP protokolu o posielanie prídavných informácií pre vzdialenú asistenciu.
3. Vytvoriť aplikáciu pre mobilné zariadenia implementujúcu navrhnutú metódu.
4. Experimentálne overiť použiteľnosť aplikácie a vyhodnotiť výsledky pri rôznych charakteristikách signálu v mobilných sieťach.
5. Vypracovať dokumentáciu podľa pokynov vedúceho práce.

Jazyk, v ktorom sa práca vypracuje: slovenský
Termín pre odovzdanie práce: 26.04.2013
Dátum zadania diplomovej práce: 31.10.2012


prof. Ing. Ján Kollár, CSc.
vedúci garantujúceho pracoviska




prof. Ing. Liberios Vokorokos, PhD.
dekan fakulty

Čestné vyhlásenie

Vyhlasujem, že som diplomovú prácu vypracoval samostatne s použitím uvedenej odbornej literatúry.

Košice 15. 4. 2013

.....

Vlastnoručný podpis

Podakovanie

Na tomto mieste sa chcem poďakovať konzultantovi a vedúcemu mojej diplomovej práce doc. Ing. Jaroslavovi Porubänovi, PhD. za pomoc, odborné vedenie, ústretový prístup a konštruktívnu kritiku pri vypracovaní práce a súvisiacich projektov.

Zároveň sa chcem poďakovať mojej rodine, priateľom a v neposlednom rade mojej priateľke Janke za trpezlivosť, pochopenie a podporu počas celého štúdia, počas vývoja projektu Mapz, ale predovšetkým počas vypracovania záverečnej práce.

Predhovor

Pokrok v oblasti informačných a komunikačných technológií (ICT) v minulých rokoch zaznamenal obrovský rast a ukázal tým jasný smer, ktorým sa budú uberať technológie v budúcnosti. Výraznou mierou prispela k tomuto faktu najmä postupná miniaturizácia výrobného procesu mobilných čipov založených prevažne na architektúre ARM a ich rapídne zvyšovanie výkonu. Inteligentné mobilné zariadenia postavené na tejto architektúre svojim výkonom a možnosťami dnes už často prevyšujúce výkon slabšieho osobného počítača, sa stali každodennou súčasťou nášho života.

Spolu s nedávnym rozmachom sietí tretej a štvrtej generácie tak dali výskumným tímom nevídané možnosti v oblasti vývoja asistívnych technológií a pomôcok pre osoby s rôznymi percepčnými poruchami zraku, sluchu a iných zmyslov. Túto skutočnosť potvrdzujú aj práce publikované v rámci niekoľkých konferencií na tému vzdialenej asistencie zrakovo postihnutých osôb [1, 2, 3], odborné články na rovnakú tému publikované v časopisoch [4, 5] a odborné články publikované v rámci výskumu na univerzitách [6].

Motiváciou pre výber tejto témy bol pre mňa predovšetkým predchádzajúci výskum v oblasti vzdialenej asistencie zrakovo postihnutých osôb, ktorý bol uskutočnený na pôde Technickej univerzity v Košiciach pod vedením doc. Ing. Jaroslava Porubäna, PhD. v rokoch 2010, 2011 a 2012 v rámci výskumného projektu Mapz [7]. Výstupom projektu Mapz bola okrem hardvérového komponentu v podobe opasku (navigačný opasok) [8] aj mobilná aplikácia s grafickým používateľským rozhraním prispôbeným pre špeciálne potreby zrakovo postihnutých osôb [9], ktorá slúži nielen na sprostredkovanie priestorových informácií nevidiacemu, ale aj na vzdialenú asistenciu nevidiaceho asistentom [10], ktorý nemusí byť na mieste fyzicky prítomný. Prepojením mobilnej aplikácie s navigačným opaskom prostredníctvom technológie bluetooth sú sprístupnené dodatočné informácie o vzdialenosti okolitých predmetov a to vďaka ultrazvukovým senzorom upevneným na navigačnom opasku. Tieto

informácie sú prístupné asistentovi v reálnom čase spolu s obrazom z kamery inteligentného mobilného telefónu, upevneného na hrudníku zrakovo postihnutej osoby. Špeciálnou funkciou, ktorej sme počas výskumu venovali najväčšiu pozornosť, sú navigačné povely zadávané asistentom na diaľku, ktoré sa na strane zrakovo postihnutej osoby prejavujú vibrovaním na tej strane opasku, na ktorú bol zaslaný navigačný povel. Navigačný opasok však dokáže fungovať aj v autonómnom režime, kedy podáva informácie o okolitých prekážkach nevidiacemu aj mimo režimu vzdialenej asistencie.

S projektom sme sa v roku 2011 zúčastnili súťaže Microsoft Imagine Cup 2011, vďaka ktorej projekt vznikol, pričom sme sa s projektom dostali vo svetovom finále, ktoré sa konalo v New Yorku (USA), do osemnástky najlepších tímov na svete. Projekt následne vyhral prvé miesto na súťaži StartupAwards.sk 2012 v kategórii Society, následkom čoho bolo označenie tvorcov projektu za jedných z tridsiatky nádeji slovenského biznisu v tohtoročnom februárovom čísle magazínu Forbes.

Obsah

Úvod	1
1 Prenos údajov v systéme vzdialenej asistencie Mapz	3
2 Formulácia úlohy	8
3 Analýza protokolov rodiny SIP	9
3.1 Základné poznatky	9
3.1.1 Účastníci SIP siete	10
3.1.2 Architektúra protokolu	11
3.1.3 Formát správy	11
3.1.4 Typy správ	12
3.1.5 Podporované typy požiadaviek	13
3.1.6 Podporované typy odpovedí	14
3.2 Súvisiace protokoly	15
3.2.1 Protokol SDP	16
3.2.2 Protokol RTP	19
3.2.3 Protokol RTCP	22
3.2.4 Ostatné protokoly	25
3.3 Ukážka komunikácie	25
3.4 Problémy protokolu SIP	32
3.4.1 Firewall	32
3.4.2 NAT (Network Address Translation)	33
3.4.3 Šifrovanie	34
3.4.4 Zložitosť	35
4 Návrh metódy prenosu údajov vzdialenej asistencie	37
4.1 Prenos pomocou protokolu MSRP	38
4.2 Rozšírenia protokolu pre kódovanie multimediálnych údajov	40

4.3	Rozšírenia hlavičky protokolu RTP	41
4.4	Analýza existujúcich implementácií	46
4.4.1	Serverové aplikácie	46
4.4.2	Aplikačné rámce	47
4.4.3	Klientské aplikácie	49
4.4.4	Ostatné aplikácie	49
5	Implementácia navrhovaného riešenia	51
5.1	Architektúra riešenia vzdialenej asistencie	52
5.2	Použité technológie	54
5.3	Prenos prídavných informácií pre účely vzdialenej asistencie	55
5.4	Rozšírenia aplikačného rámca doubango	57
5.4.1	Signalizácia podporovaných rozšírení hlavičky RTP	57
5.4.2	Prenos dohodnutých rozšírení v RTP paketoch	59
5.5	Mobilná aplikácia pre vzdialenú asistenciu	61
5.5.1	Používateľské prostredie a spôsob fungovania	61
5.5.2	Odosielanie a prijímanie prídavných informácií počas hovoru	62
6	Vyhodnotenie použiteľnosti riešenia	66
6.1	Metodológia testovania	66
6.2	Vyhodnotenie doby oneskorenia obrazu	67
6.3	Vyhodnotenie objemu prenášaných obrazových údajov	67
6.4	Celkové hodnotenie riešenia	68
7	Záver	69
	Zoznam použitej literatúry	70
	Zoznam príloh	77
8	Používateľská príručka	79

8.1	Funkcia aplikácie	79
8.2	Súpis obsahu dodávky	79
8.3	Hardvérové a softvérové požiadavky	80
8.4	Inštalácia aplikácie	81
8.5	Použitie aplikácie	81
8.5.1	Prihlásenie sa do aplikácie	83
8.5.2	Nadviazanie VoIP spojenia pre účely vzdialenej asistencie . . .	84
8.5.3	Prepnutie účtov a odhlásenie sa z aplikácie	85
8.6	Chybové hlásenia	86
8.7	Obmedzenia	88
9	Systémová príručka	90
9.1	Analýza riešenia	90
9.2	Architektúra	91
9.3	Opis natívnej knižnice	92
9.3.1	Preklad natívnej knižnice	95
9.4	Opis knižnice android-ngn-stack	95
9.5	Opis mobilnej aplikácie	97
9.5.1	Preklad aplikácie	98
9.6	Vývojové nástroje a knižnice tretích strán	99

Zoznam obrázkov

1–1	Architektúra súčasného riešenia vzdialenej asistencie v projekte Mapz.	4
1–2	Využitie prídavných informácií o polohe a natočení nevidiaceho v súčasnom riešení vzdialenej asistencie Mapz. [10]	6
3–1	Ukážka priamej P2P komunikácie.	10
3–2	Ukážka komunikácie za pomoci proxy servera.	10
3–3	Vzťah protokolov SIP a RTP v prebiehajúcom sedení.	22
3–4	Ukážka nadviazania hovoru pomocou protokolu SIP za použitia proxy servera. [15]	26
4–1	Protokoly rodiny SIP vo vzťahu k protokolom nižších vrstiev a k UA.	38
5–1	Architektúra implementovaného riešenia vzdialenej asistencie Mapz.	53
5–2	Pohľad na balíky aplikačného rámca doubango. [43]	56
8–1	Inštalácia aplikácie <i>Vzdialený asistent</i> .	82
8–2	Spustenie aplikácie a prihlásenie sa	84
8–3	Pokus o nadviazanie VoIP spojenia medzi nevidiacim a asistentom.	85
8–4	Prebiehajúce VoIP sedenie medzi nevidiacim a asistentom.	86
8–5	Odhlásenie sa a prepnutie účtov.	87
8–6	Chybové hlásenia v aplikácii <i>Vzdialený asistent</i> .	87
8–7	Problémy s obrazom v prípade nekvalitného pripojenia.	88
9–1	Časti implementovaného systému.	91
9–2	Architektúra aplikačného rámca doubango.	92
9–3	Stiahnutie SDK platformy Android 4.2.2.	98
9–4	Kontrola väzieb medzi projektmi v prostredí Eclipse.	99

Zoznam tabuliek

3–1	Podporované typy požiadaviek protokolu SIP	13
3–2	Ukážky a význam stavových kódov odpovedí protokolu SIP.	15
3–3	Povolené hodnoty položky <typ> v session-level sekcii vo formáte protokolu SDP.	18
3–4	Povolené hodnoty položky <typ> v media-level sekcii vo formáte pro- tokolu SDP.	18

Zoznam symbolov a skratiek

3G 3rd generation

3GPP 3rd Generation Partnership Project

4G 4th generation

ASCII American Standard Code for Information Interchange

GPS Global Positioning System

GSM Global System for Mobile Communications

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

ICE Interactive Connectivity Establishment

IETF Internet Engineering Task Force

IP Internet Protocol

IPv4 Internet Protocol version 4

IPv6 Internet Protocol version 6

MJPEG Motion JPEG

NAT Network Address Translator/Translation

NTP Network Time Protocol

OSI **O**pen **S**ystems **I**nterconnection

P2P **P**eer-**to**-**P**eer

PKI **P**ublic **K**ey **I**nfrastructure

REST **R**epresentational **S**tate **T**ransfer

RFC **R**equest **F**or **C**omments

RTCP **R**eal-time **T**ransport **C**ontrol **P**rotocol

RTP **R**eal-time **T**ransport **P**rotocol

SDP **S**ession **D**escription **P**rotocol

SIP **S**ession **I**nitiation **P**rotocol

SMTP **S**imple **M**ail **T**ransfer **P**rotocol

SRTP **S**ecure **R**eal-time **T**ransport **P**rotocol

STUN **S**ession **T**raversal **U**tilities for **N**AT

TCP **T**ransmission **C**ontrol **P**rotocol

TLS **T**ransport **L**ayer **S**ecurity

TURN **T**raversal **U**sing **R**elays around **N**AT

UA **U**ser **A**gent

UAC **U**ser **A**gent **C**lient

UAS **U**ser **A**gent **S**erver

UDP **U**ser **D**atagram **P**rotocol

URI **U**nified **R**esource **I**dentifier

UTF-8 **U**CS **T**ransformation **F**ormat - **8**-bit

VoIP **V**oice **o**ver **I**P

XML **E**xtensible **M**arkup **L**anguage

Slovník termínov

ARM je architektúra opisujúca rodinu RISC počítačových procesorov, navrhnutá britskou spoločnosťou ARM Holdings.

Cloud Computing je využívanie výpočtových prostriedkov (hardvéru a softvéru), ktoré sú poskytované ako služba cez internetovú sieť.

Enkódovanie / Dekódovanie je proces prípravy nespracovaného multimediálneho obsahu (napríklad zvuk alebo video) pre vytvorenie výstupného súboru, ktorý spĺňa podmienky požadovaného formátu.

Firewall je softvér alebo hardvérové zariadenie, ktoré slúži na zabezpečenie siete. Filtrovanie paketov sa vykonáva na základe ich analýzy.

Global Positioning System je systém satelitnej navigácie, ktorý poskytuje informácie o polohe a čase za každých poveternostných podmienok.

GSM hovor je klasický telefónny hovor v celulárnej sieti 2. generácie (GSM).

Interactive Connectivity Establishment je technika používaná v počítačových sieťach, ktoré obsahujú zariadenia NAT na obchádzanie obmedzení, ktoré plynú z použitia týchto zariadení.

Internet Engineering Task Force je veľká internetová komunita sieťových návrhárov, výrobcov a výskumných pracovníkov, ktorí sa zaoberajú evolúciou internetovej architektúry.

Network Address Translation je proces zmeny informácie o IP adrese v hlavičkách IPv4 paketov pri prechode smerovačmi internetovej komunikácie.

Paket je formátovaný blok údajov prenášaný v paketovo orientovanej sieti.

Protokoly rodiny SIP sú všetky protokoly, ktoré sú potrebné pre správne fungo-

vanie technológie SIP vrátane protokolov, ktoré len rozširujú základné funkcionality tejto technológie.

Public Key infraštruktúra je množina hardvéru, softvéru, ľudí, pravidiel a postupov, potrebných pre vytváranie, správu, distribúciu, používanie, uchovávanie a odvolávanie digitálnych certifikátov.

Referenčná implementácia je názorná implementácia, štandardne dodávaná ku knižniciam zdrojového kódu, ktorá poukazuje na základné postupy použitia danej knižnice.

Representational State Transfer je štýl softvérovej architektúry, navrhnutý pre distribuované systémy.

Senzorické údaje sú informácie, získané z rôznych senzorov daného zariadenia.

Serializácia / Deserializácia je proces transformácie údajových štruktúr a objektov do formátu, ktorý môže byť použitý pre uchovávanie a prenos týchto informácií. Opakom serializácie je deserializácia, pomocou ktorej sa údaje z daného formátu transformujú späť do podoby údajových štruktúr a objektov.

Siete 3. generácie sú telekomunikačné siete využívajúce technológie 3. generácie ako napríklad 3G, HSDPA a HSUPA.

Siete 4. generácie sú telekomunikačné siete využívajúce technológie 4. generácie ako napríklad LTE.

Voice over IP je technológia, ktorá sa zaoberá prenosom zvukových údajov medzi dvoma a viacerými klientmi cez IP sieť, akou je napríklad sieť internet, pričom klientom môže byť ľubovoľné zariadenie s mikrofónom a reproduktorom.

Wi-Fi je technológia, ktorá umožňuje bezdrôtové spojenie elektronických zariadení v počítačovej sieti, založené na šírení rádiových vĺn.

Úvod

Technológia VoIP ako taká existuje už približne od roku 1995. K jej masovému rozšíreniu však došlo až po roku 2000 s príchodom dostupného širokopásmového internetu, v tých časoch prevažne v západných krajinách, ktorý zohrával veľmi podstatnú úlohu v jej šírení. Po prvýkrát v histórii internetu tak mali ľudia k dispozícii dostatočnú šírku pásma na používanie kodekov vyššej kvality a uskutočňovanie VoIP hovorov súbežne s používaním internetových prehliadačov a iných programov používajúcich internetové pripojenie bez toho, aby dochádzalo k jeho nadmernému zahľteniu. Niektoré firmy ako Cisco a Lucent si v tých rokoch na službách VoIP dokonca postavili svoj biznis.

V počiatočných rokoch dostupnosti technológie VoIP vzniklo zároveň niekoľko štandardov pre hlasovú komunikáciu v IP sieti. Jedným z mnohých je aj protokol SIP, ktorý pôvodne navrhli Henning Schulzrinne a Mark Handley v roku 1996. Tento protokol bol v roku 2000 prijatý ako 3GPP signalizačný protokol. Za projektom 3GPP v súčasnosti stojí šesť organizácií z celého sveta, orientujúcich sa na vývoj telekomunikačných štandardov. Protokol prešiel od svojho prvotného uvedenia mnohými zmenami a v súčasnej podobe z júna roku 2002 je prístupný ako RFC štandard udržiavaný komunitou IETF.

Samotný protokol SIP však pri poskytovaní VoIP služieb zodpovedá len za signalizáciu a sprostredkovanie spojenia. Nezodpovedá za popis prenášaných zvukových a obrazových údajov, ani za ich prenos. Na tento účel využíva služby protokolov SDP a RTP, ktorými sa vo veľkej miere táto práca zaoberá. Obidva protokoly sú tiež štandardizované komunitou IETF ako RFC štandardy, o čo sa táto práca výrazne opiera. Fungovanie a štruktúra protokolu SIP a ním využívaných protokolov bude bližšie opísaná v kapitole 3.

Účelom tejto práce je nájsť vhodné riešenie pre rozšírenie služieb protokolu SIP,

prípadne protokolov SDP a RTP o možnosť posielania rozširujúcich údajov, potrebných pre poskytovanie priestorových informácií vzdialenému asistentovi v zmysle požiadaviek projektu Mapz. Toto riešenie by malo využívať iba existujúce možnosti štruktúry spomínaných protokolov a držať sa predpísaných štandardov, aby sa predišlo neskorším problémom s kompatibilitou riešenia. Možnosti rozšírenia protokolov budú analyzované v kapitole 3, na ktorú plynule nadviaže kapitola 4, pojednávajúca o návrhu metódy rozšírenia protokolov.

Výstupom práce by mal byť rozšírený aplikačný rámec poskytujúci funkcionality VoIP klienta, ktorého funkčnosť bude demonštrovaná na vyvinutej mobilnej aplikácii. Implementácia rozšírení aplikačného rámca a mobilnej aplikácie bude opísaná v kapitole 5. Použitelnosť tohto riešenia a prípadné návrhy na ďalší postup budú uvedené v kapitolách 6 a 7.

1 Prenos údajov v systéme vzdialenej asistencie

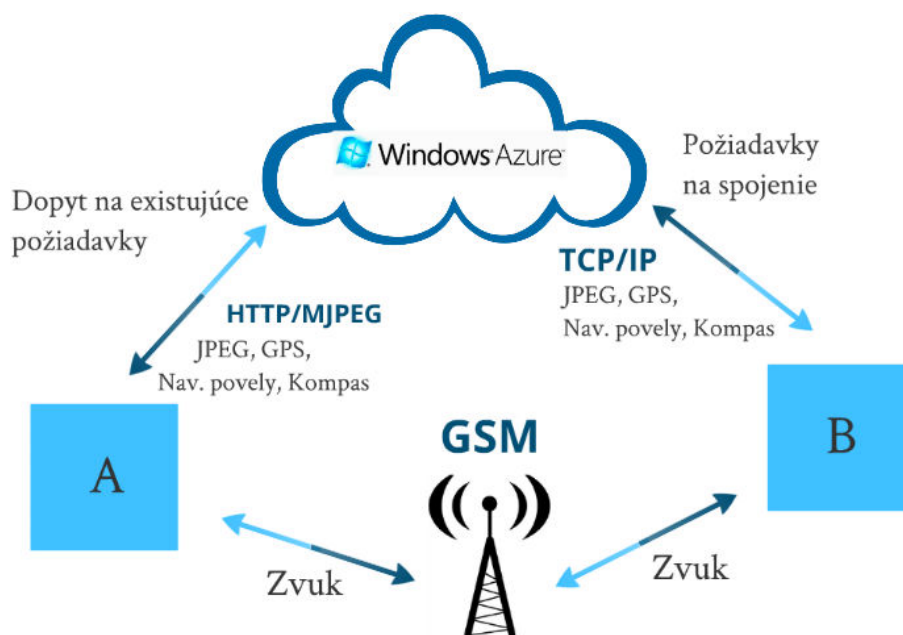
Mapz

Predtým, než sa dostaneme k analýze štruktúry a možností rozšírenia protokolov rodiny SIP, je potrebné pre neskoršie porovnanie uviesť, v akom stave sa nachádza implementácia funkcionality vzdialenej asistencie v projekte Mapz v súčasnosti. Konkrétne pre účely tejto práce uvediem, akým spôsobom sú počas vzdialenej asistencie prenášané zvukové údaje, obrazové údaje a prídavné informácie ako sú napríklad GPS súradnice, informácie získané z digitálneho kompasu a navigačné povely medzi dvoma rôznymi mobilnými aplikáciami, z ktorých je jedna určená pre nevidiaceho a druhá pre asistenta.

Na obrázku 1–1 je v jednoduchosti znázornená súčasná architektúra vzdialenej asistencie v projekte Mapz. Predpokladajme, že A je vzdialený asistent a B je nevidiaci. Nevidiaci požiadava o vzdialenú asistenciu zaslaním požiadavky prostredníctvom mobilnej aplikácie Mapz. Požiadavka na vzdialenú asistenciu sa odosiela cez webové služby na server, ktorý je umiestnený v infraštruktúre Windows Azure. Zároveň v tej istej chvíli mobilná aplikácia nevidiaceho vytáča telefónne číslo asistenta v GSM sieti. Mobilná aplikácia asistenta periodicky volá tie isté webové služby a dopytuje sa, či nedorazila požiadavka na vzdialenú asistenciu smerujúca prihlásenému asistentovi, čím sa získava ID nadchádzajúceho sedenia. Po zodvihnutí telefónneho hovoru asistentom sa nadviaže komunikačný kanál cez proxy server umiestnený v tej istej infraštruktúre pomocou získaného ID sedenia. Kvôli jednoduchosti sme sa počas vývoja nezaoberali možnosťou priameho P2P spojenia medzi zúčastnenými stranami, takže celá komunikácia prebieha cez spomínaný proxy server.

Po úspešnom nadviazaní spojenia oboma stranami s proxy serverom, mobilná aplikácia nevidiaceho začneodosielať snímky vo formáte JPEG spolu s GPS súradnicami a ďalšími senzorickými údajmi prostredníctvom špeciálne navrhnutého pro-

tokolu, fungujúceho v aplikačnej vrstve nad protokolom TCP/IP. Na druhej strane sa na prenos snímok z proxy servera do mobilnej aplikácie asistenta využíva technológia Motion JPEG (MJPEG) [11], čo je vlastne tok statických JPEG snímok, získaných z kamery mobilného zariadenia. Snímky sa prenášajú cez nadviazaný komunikačný kanál v tele nekonečnej HTTP odpovede. Protokol HTTP bol zvolený kvôli tomu, že to bol v tom čase jediný podporovaný komunikačný kanál na platforme Windows Phone 7, na ktorej bola postavená mobilná aplikácia pre asistenta. Platforma Windows Phone 7 vtedy ešte podporu pre TCP a UDP spojenia vo verejne prístupnom oficiálnom aplikačnom rozhraní neobsahovala.



Obr. 1 – 1 Architektúra súčasného riešenia vzdialenej asistencie v projekte Mapz.

Pre lepšie pochopenie použitej technológie MJPEG a nekonečného HTTP toku údajov je nižšie uvedený výpis takejto HTTP odpovede. Úvodné riadky obsahujú základné HTTP hlavičky, odosielané HTTP serverom, pričom za zmienu stojí hodnota použitá v hlavičke *Content-Type*. Táto hodnota obsahuje okrem typu prenášaných údajov (*multipart/x-mixed-replace*) aj časť s názvom *boundary*, ktorej hodnota určuje oddeľovač jednotlivých JPEG snímok v tele nekonečnej HTTP odpovede. Hneď za hlavičkami, v tele HTTP odpovede nasleduje prvý oddeľovač snímok (hodnota “–

myboundary”), na označenie toho, že ďalej nasleduje sekcia, v ktorej sa nachádzajú informácie o snímke. Tieto informácie sa uvádzajú ako hodnoty HTTP hlavičiek prislúchajúcich prvej snímke, ktorá je uvedená pod nimi. Za povinné sa považujú hlavičky *Content-Type*, ktorá obsahuje formát snímky a *Content-Length*, ktorá obsahuje veľkosť snímky v bajtoch. Hlavička *X-Data* obsahuje prídavné informácie potrebné pre zefektívnenie vzdialenej asistencie. Jedná sa predovšetkým o GPS súradnice a informáciu o natočení nevidiaceho. Ukážka využitia týchto informácií je na obrázku 1–2. Tieto informácie slúžia na označenie polohy a natočenia nevidiaceho na mape, ktorá je zobrazená v hornej polovici displeja nad obrazom z kamery nevidiaceho. Za hlavičkami sa nachádzajú už len binárne údaje snímky vo formáte, ktorý určuje hlavička *Content-Type*, v našom prípade sa používa formát JPEG. Snímky sú takto naskladané za sebou a oddelené hodnotou “*–myboundary*”.

```
HTTP/1.1 200 OK
Content-Type: multipart/x-mixed-replace; boundary=--myboundary
Server: Microsoft-IIS/8.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Fri, 12 Apr 2013 15:48:27 GMT
...
--myboundary
X-Data: <zemepisna-dlзка-1>;<zemepisna-sirka-1>;<natocenie-1>
Content-Type: image/jpeg
Content-Length: <velkost-snimky-1>

<binarne-udaje-snimky-1>
--myboundary
X-Data: <zemepisna-dlзка-2>;<zemepisna-sirka-2>;<natocenie-2>
Content-Type: image/jpeg
Content-Length: <velkost-snimky-2>

<binarne-udaje-snimky-2>
...
```

Jedným z výrazných nedostatkov systému vzdialenej asistencie v projekte Mapz je jeho ekonomická nerentabilita. Na posielanie obrazových údajov od zrakovo pos-



Obr. 1–2 Využitie prídavných informácií o polohe a natočení nevidiaceho v súčasnom riešení vzdialenej asistencie Mapz. [10]

tihnutej osoby k asistentovi sa používa technológia Motion JPEG (MJPEG). Binárne údaje snímok sú prenášané v nekonečnej HTTP odpovedi. Riešenie navyše využíva finančne náročnú cloudovú infraštruktúru Windows Azure ako proxy server pre prenos údajov. Ak vezmeme do úvahy, že každá JPEG snímka s rozmermi 355x288 pixelov má veľkosť 4-6 kilobajtov, prídavné informácie sú prenášané v textovom formáte s kódovaním UTF-8 (s variabilnou dĺžkou jedného znaku v bajtoch) a všetko je zabalené do HTTP odpovede, zistíme, že náklady za objem prenesených údajov by boli nesmierne vysoké. Okrem toho je oneskorenie prichádzajúcich snímok a celková plynulosť súčasnej verzie videoprenosu počas vzdialenej asistencie na veľmi nízkej úrovni, predovšetkým kvôli absencii mechanizmu kontroly kvality. Pre účely zvukového hovoru sa na druhej strane využíva spoplatnený telefónny hovor v GSM sieti, čo znamená ďalšie náklady navyše. Vďaka týmto a ďalším problémom je toto riešenie po vykonaných praktických testoch s nevidiacimi v rámci mobilnej siete takmer nepoužiteľné.

Ambíciou nášho tímu, ktorý stojí za projektom Mapz, je teda nahradiť súčasné riešenie vzdialenej asistencie novým riešením, založeným na technológii SIP. Tá bola uprednostnená pred ostatnými VoIP technológiami na základe rozhovorov s členmi laboratória počítačových sietí (CNL) na Technickej Univerzite v Košiciach. Členovia CNL nám poskytli cenné rady, odporúčania a zároveň prisľúbili svoju odbornú pomoc v oblasti VoIP technológií počas vývoja projektu Mapz.

2 Formulácia úlohy

Cielom tejto práce je nájsť nový, efektívny a štandardizovaný spôsob posielania prídavných informácií zabalených priamo do multimediálneho toku údajov kvôli efektivite a potrebe synchronizácie niektorých informácií s jednotlivými obrazovými snímkami. Pri návrhu metódy rozšírenia protokolov je dôležité dbať na kompatibilitu riešenia s modernou infraštruktúrou verejne prístupných SIP serverov a držať sa, ak je to možné, predpísaných štandardov. Zdrojom informácií vo fáze návrhu budú predovšetkým RFC štandardy publikované komunitou IETF. Na konci práce je potrebné objektívne zhodnotiť výhody a nevýhody technológie SIP použitej namiesto súčasného riešenia vzdialenej asistencie Mapz v mobilných sieťach, pričom pre porovnanie kvalitatívnych parametrov budú použité výsledky dosiahnuté súčasným riešením vzdialenej asistencie Mapz.

Pre dosiahnutie uvedeného cieľa je potrebné vykonať nasledujúce kroky:

- Oboznámiť sa so štruktúrou a fungovaním protokolov rodiny SIP.
- Analyzovať možnosti rozšírenia protokolov o posielanie prídavných informácií.
- Navrhnuť metódu rozšírenia protokolov pre účely posielania priestorových informácií.
- Analyzovať dostupné aplikačné rámce s podporou protokolu SIP.
- Implementovať navrhnutú metódu do vybraného aplikačného rámca, poskytujúceho služby VoIP klienta.
- Implementovať mobilnú aplikáciu využívajúcu rozšírený aplikačný rámec pre overenie použiteľnosti riešenia.
- Vyhodnotiť použiteľnosť vytvoreného riešenia vo vzťahu k súčasnému stavu.

3 Analýza protokolov rodiny SIP

Protokol SIP je podľa [12] jedným zo štandardizovaných VoIP protokolov, v súčasnosti udržiavaný komunitou IETF. Aktuálna verzia protokolu je *SIP/2.0* a je opísaná štandardom RFC 3261 [13], z ktorého budem čerpať väčšinu informácií.

SIP je vo veľkej miere využívaný v rámci IP multimediálnych subsystémov (IMS), ktoré poskytujú multimediálne služby v mobilných telefónoch tretej (3G) a štvrtej generácie (4G), čím je zabezpečená priemyselná podpora od výrobcov. Tento fakt nasvedčuje tomu, že protokol bude v nadchádzajúcich rokoch stále relevantný.

Rozsiahlosť protokolu SIP a s ním súvisiacich protokolov nedovoľuje opísať na nasledujúcich stranách všetky jeho možnosti a preto sa budem sústrediť na základný koncept fungovania, prepojenia protokolov a možností, ktoré vo svojej kombinácii poskytujú pre účely neskoršej analýzy možných rozšírení.

3.1 Základné poznatky

Ako už bolo v úvode tejto práce spomenuté, SIP je signalizačný protokol, ktorý zodpovedá len za nadviazanie, koordináciu a ukončenie VoIP sedenia medzi dvoma koncovými bodmi v paketovo orientovanej sieti. Za popis a samotnú výmenu multimediálnych údajov zodpovedajú iné protokoly, ktoré budú opísané neskôr.

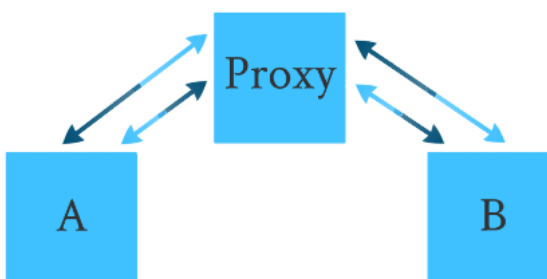
Z pohľadu komunikácie je protokol navrhnutý ako P2P protokol, čo znamená, že všetky prítomné strany môžu v komunikácii zohrávať ako úlohu servera, tak aj úlohu klienta, na základe toho, kto je odosielateľ (klient) a kto prijímateľ danej požiadavky (server). Komunikácia môže teda prebiehať priamo. Ukážka takejto komunikácie je znázornená na obrázku 3–1.

V praxi však väčšina implementácií obsahuje aj proxy server, kvôli obmedze-



Obr. 3–1 Ukážka priamej P2P komunikácie.

niam priameho spojenia, ktoré budú opísané v podkapitole 3.4. Tento typ komunikácie je na obrázku 3–2.



Obr. 3–2 Ukážka komunikácie za pomoci proxy servera.

3.1.1 Účastníci SIP siete

Špecifikácia protokolu SIP rozlišuje štyri základné typy logických entít, ktoré sa zúčastňujú komunikácie a sú opísané v [15], kapitola 4. Jedná sa o:

User Agent (UA), ktorý je koncovým bodom v sieti. UA nadväzuje a ukončuje sedenie výmenou požiadaviek a odpovedí s iným UA. Špecifikácia protokolu [13] ďalej rozlišuje niekoľko typov UA:

- **User Agent Client (UAC)** - klientská aplikácia, ktorá zahajuje SIP požiadavky.
- **User Agent Server (UAS)** - serverová aplikácia, ktorá prijíma SIP požiadavky a odosiela na ne odpoveď.

Redirect Server (RS), ktorý pomáha lokalizovať UA poskytovaním alternatív-

nych adries, na ktorých môže byť používateľ zastihnutý. V ďalšej komunikácii sa na zastihnutie UA namiesto starej adresy používa nová adresa, ktorú vrátil Redirect Server. Tento server nepreposiela prijaté požiadavky priamo.

Proxy Server (PS), ktorý zohráva dvojité úlohu servera a klienta na účely odsielania požiadaviek v mene iných klientov. Prijaté požiadavky sú spracované buď interne alebo sa preposielajú ďalej na iné servery. Tento server interpretuje a v prípade nutnosti prepisuje požiadavky pred ich preposlaním. Na rozdiel od RS, týmto serverom prechádza komunikácia za každých okolností.

Registrar, ktorý je zodpovedný za prijímanie REGISTER požiadaviek. Na základe týchto požiadaviek sa upravuje interná databáza adries, na ktorých je používateľ dostupný.

3.1.2 Architektúra protokolu

Z pohľadu architektúry je SIP textovo orientovaný protokol aplikačnej vrstvy, založený na transakčnom modele požiadavka-odpoveď. Najlepšie môžeme protokol SIP opísať ako hybrid medzi protokolom SMTP [16] a protokolom HTTP [17]. Transakčný model fungovania a základná štruktúra sú prebraté od protokolu HTTP a každý SIP hovor smeruje od niekoho (hlavička *From*), niekomu (hlavička *To*), podobne ako u protokolu SMTP. Na druhej strane, protokol okrem základnej štruktúry preberá od protokolu HTTP aj niektoré stavové kódy, ktoré budú opísané neskôr. SIP však aj napriek podobnostiam nie je rozšírením protokolu HTTP, ani SMTP.

3.1.3 Formát správy

Typická SIP správa je kódovaná v UTF-8. Každá SIP správa obsahuje podobne ako správa protokolu HTTP tieto časti:

Úvodný riadok, ktorý určuje typ správy a jeho podoba je na tomto type závislá.

Typy správ sú uvedené v nasledujúcej podkapitole.

Hlavičky, ktoré sú podobne ako pri protokole HTTP uvedené každá na samostatný riadok, oddelené znakom konca riadku. Na každom riadku je znakom dvojbodky oddelený názov hlavičky a hodnota hlavičky vo formáte:

<názov>:<hodnota>

Telo správy, ktoré sa používa na opísanie sedenia, ktoré bude prebiehať. V tele môžu byť uvedené použité audio a video kodeky, vzorkovacie frekvencie a podobne. Telo správy sa väčšinou uvádza v protokole s názvom SDP, ktorému je v práci venovaná samostatná stať.

3.1.4 Typy správ

V štandarde protokolu SIP sú na základe použitého transakčného modelu požiadavka-odpoveď opísané dva základné typy správ, ktorých úvodný riadok určuje, o ktorý z nasledujúcich typov správy sa jedná:

Požiadavky, ktorých úvodný riadok obsahuje podobne ako požiadavka protokolu HTTP volanú metódu, adresu (štandardizovaná SIP adresa) a verziu protokolu (aktuálne *SIP/2.0*) nasledovanú znakom konca riadku. Typická požiadavka vyzerá podobne ako na nasledujúcom výpise:

```
INVITE sip:tstanik@mapzproject.org SIP/2.0
Via: SIP/2.0/UDP 10.0.0.7:34856;branch=z9hG4bK-[...]--d87543-;rport
Max-Forwards: 70
Contact: <sip:msuscak@10.0.0.7:34856>
To: "tstanik"<sip:tstanik@mapzproject.org>
From: "msuscak"<sip:msuscak@mapzproject.org>;tag=7c32af6e
Call-ID: 0TFiNjc0YjMzZDg1Y2Q3ZGVhZDIxOTY1OGRmMTg5MzU.
CSeq: 1 INVITE
...
```

Odpovede, ktorých úvodný riadok obsahuje podobne ako odpoveď protokolu HTTP verziu protokolu (opäť *SIP/2.0*), stavový kód odpovede a krátky popisný text stavového kódu nasledovaného znakom konca riadku. Typická odpoveď vyzerá podobne ako na nasledujúcom výpise:

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 10.0.0.7:34856;received=10.1.2.69;branch=z9[...]-43-;rport
    =34856
Record-Route: <sip:10.1.2.48;lr;pm;n1>
Contact: <sip:tstanik@10.1.2.69:1026;rinstance=0ca247682fe6140e>
To: "msuscak"<sip:msuscak@mapzproject.org>;tag=673dc94f
From: "tstanik"<sip:tstanik@mapzproject.org>;tag=7c32af6e
Call-ID: 0TFiNjc0YjMzZDg1Y2Q3ZGVhZDIxOTY1OGRmMTg5MzU.
CSeq: 2 INVITE
...
```

3.1.5 Podporované typy požiadaviek

Špecifikácia protokolu SIP rozlišuje niekoľko typov požiadaviek na základe metódy uvedenej v úvodnom riadku SIP požiadavky, ktoré sú uvedené v tabuľke 3 – 1.

Tabuľka 3 – 1 Podporované typy požiadaviek protokolu SIP

Metóda	Popis použitia
INVITE	Nadviazanie SIP sedenia a zmena parametrov hovoru (re-INVITE).
ACK	Potvrdenie prijatia odpovede na INVITE požiadavku.
BYE	Ukončenie hovoru.
CANCEL	Zrušenie požiadavky na nadviazanie spojenia a “vyzváňania”.
OPTIONS	Zistenie možností klienta na druhej strane spojenia.
REGISTER	Registrácia UA v lokalizačnej službe.
INFO	Posielanie informácií o sedení uprostred prebiehajúceho hovoru, bez zmeny stavu hovoru.

3.1.6 Podporované typy odpovedí

Na základe stavového kódu odpovede rozlišujeme dva základné typy SIP odpovedí. Stavové kódy sa zoskupujú do takzvaných tried stavových kódov. Stavový kód sa skladá z troch čísiel a hovorí o výsledku spracovania prijatej požiadavky a trieda stavových kódov je skupina stavových kódov s podobnými vlastnosťami. Medzi základné typy odpovedí patria:

Dočasné odpovede, ktoré server používa na indikáciu priebehu rôznych akcií.

Tieto odpovede neukončujú SIP transakciu. Patrí sem len jedna trieda stavových kódov:

- **1xx** - Používa sa na informovanie o úspešnom prijatí požiadavky a stále prebiehajúcim spracovaní.

Koncové odpovede, ktoré na rozdiel od dočasných ukončujú SIP transakciu.

Patria sem nasledujúce triedy stavových kódov:

- **2xx** - Používa sa na informovanie o úspešnosti prijatia požiadavky a spracovania transakcie.
- **3xx** - Používa sa na informovanie o potrebe presmerovania, prípadne inej vykonanej akcie pre účely dokončenia spracovania.
- **4xx** - Používa sa na informovanie o chybách v došlej požiadavke alebo nemožnosti spracovania.
- **5xx** - Používa sa na informovanie o zlyhaniach na serveri aj napriek korektnej požiadavke.
- **6xx** - Používa sa na informovanie o globálnych zlyhaniach ako sú zaneprázdnenosť klienta, odmietnutie alebo nedostupnosť pri pokuse o nadviazanie spojenia.

Ako môžeme vidieť, protokol SIP naozaj preberá množstvo vlastností tried stavových kódov od protokolu HTTP. Ukážky a význam niektorých základných stavových kódov odpovedí protokolu SIP sú uvedené v tabuľke 3–2.

Tabuľka 3–2 Ukážky a význam stavových kódov odpovedí protokolu SIP.

Stavový kód	Význam	Stavový kód	Význam
100	Pokračovanie.	408	Vypršal časový limit požiadavky.
180	Vyzváňanie.	480	Nedostupný.
200	OK.	481	Transakcia neexistuje.
300	Viac možností.	482	Zistená slučka.
301	Presunuté permanentne.	5xx	Špecifické chyby na serveri.
302	Presunuté dočasne.	600	Zaneprázdnený.
400	Neplatná požiadavka.	603	Odmietnutá požiadavka.
401	Neautorizovaný prístup.	604	Neexistuje.
403	Zakázaný prístup.	606	Požiadavka nemôže byť prijatá.

3.2 Súvisiace protokoly

Ako už bolo uvedené v predchádzajúcich kapitolách, protokol SIP zabezpečuje len úkony spojené so signalizáciou, čo znamená nadviazanie, koordináciu a ukončenie spojenia. Nezabezpečuje však samotný prenos multimedialných údajov. Okrem toho pre účely poskytovania kvalitných hlasových a obrazových služieb je potrebné mať mechanizmus kontroly kvality a vzhľadom k tomu, že existuje množstvo zvukových a obrazových kodekov, je zároveň potrebné vopred dohodnúť, ktoré kodeky sa budú počas multimedialného sedenia používať. Samozrejme, bolo by možné nájsť oveľa viac atribútov a procesov dôležitých pre VoIP služby, o ktorých špecifikácia proto-

kolu SIP nič nehovorí. Protokol bol totiž už od jeho počiatkov navrhovaný s cieľom vedúcim k integrácii existujúcich otvorených štandardov, ktorých spojením by mohol vzniknúť jeden dobre fungujúci celok.

Pre úplné pochopenie základných konceptov fungovania celého SIP protokolu v zmysle poskytovania VoIP služieb je preto potrebné si ešte objasniť úlohu protokolov SDP, RTP a RTCP, ktoré zodpovedajú za:

- Popis multimediálneho sedenia z pohľadu použitých kodekov a kvalitatívnych parametrov (protokol SDP).
- Prenos multimediálneho obsahu, predovšetkým zvukových a obrazových údajov (protokol RTP).
- Kontrolu a variabilné prispôsobovanie kvalitatívnych parametrov prenášaných údajov (protokol RTCP).

Fungovanie, štruktúru a možnosti jednotlivých protokolov uvediem v nasledujúcich statiach.

3.2.1 Protokol SDP

Pri nadväzovaní multimediálnych telekonferencií a hlasových hovorov v IP sieťach je potrebné nejakým spôsobom medzi dvoma a viacerými koncovými bodmi dohodnúť typové a kvalitatívne detaily prenášaných údajov. Môže sa jednať o zvuk, obraz alebo iný typ prenášaných údajov medzi účastníkmi.

Protokol SDP poskytuje štandardizovanú reprezentáciu takýchto informácií, bez ohľadu na to, akým spôsobom sú dané údaje prenášané. SDP je formát učný len na popis nadchádzajúceho multimediálneho sedenia. Nepojednáva o spôsobe prenosu týchto informácií a musí teda využívať iné protokoly určené pre prenos. Na prenos informácií vo formáte SDP sa preto používa telo SIP správy.

SDP bol navrhnutý ako protokol všeobecného zamerania so zámerom jeho použitia v rámci rôznych typov sieťových prostredí a rôznych typov aplikácií. Vzhľadom k jeho rozsiahlosti nebudem opisovať všetky jeho možnosti a vlastnosti, ktoré opisujú IETF štandardy RFC 4566 [18], hovoriaci o samotnom protokole SDP a RFC 3264 [19], vytvorený v súvislosti s jeho integráciou do protokolu SIP. Protokol bol podobne ako SIP rozšírený ďalšími štandardmi, ktoré sú mimo rozsah tejto práce.

Protokol SDP sa označuje typom *application/sdp* a je to textovo orientovaný protokol aplikačnej vrstvy, ktorý na kódovanie textu využíva UTF-8. Názvy položiek a atribútov však používajú len podmnožinu povolených znakov označenú ako ASCII. Na rozdiel od protokolu SIP, protokol SDP obsahuje len telo, ktoré pozostáva z množiny riadkov, kde každý riadok je uvedený vo formáte *<typ>=<hodnota>*. Položka *<typ>* musí obsahovať presne jeden znak z množiny znakov, ktoré sú závislé na veľkosti a *<hodnota>* obsahuje štrukturovaný text, ktorého formát závisí od toho, čo obsahuje *<typ>*. Vo všeobecnosti je však *<hodnota>* buď množina položiek (resp. refazcov) oddelených znakom medzery alebo refazec v ľubovoľnom tvare.

Telo protokolu SDP pozostáva z niekoľkých sekcií a vždy začína sekciou, ktorá obsahuje riadky súvisiace so sedením (takzvaná session-level sekcia, ktorá začína riadkom “*v=...*”) a za nimi nasleduje nula a viac riadkov súvisiacich s médiami (takzvané media-level sekcie, ktoré začínajú každá riadkom “*m=...*”). Niektoré povolené hodnoty položky *<typ>* v session-level sekcii sú v tabuľke 3–3, v media-level sekcii zas v tabuľke 3–4. Hodnoty v stĺpci s názvom “*” v tabuľke hovoria o tom, či sa daný typ môže v tele vyskytovať viackrát.

Množina znakov, ktoré sú závislé na veľkosti a používajú sa v hodnotách položky *<typ>* je značne obmedzená, takže neposkytuje veľa priestoru na rozšírenie protokolu použitím znakov, ktoré v špecifikácii nemajú zatiaľ priradený význam. Syntaktický analyzátor musí riadky, ktorých položka *<typ>* obsahuje znak bez v špecifikácii priradeného významu úplne ignorovať. Na účely rozširovania možností protokolu

Tabuľka 3–3 Povolené hodnoty položky <typ> v session-level sekcii vo formáte protokolu SDP.

<typ>	Význam	*
v	Verzia protokolu.	Nie
o	Pôvodca a identifikátor sedenia.	Nie
s	Názov sedenia.	Nie
c	Informácie o spojení.	Áno
a	Atribúty sedenia.	Áno

Tabuľka 3–4 Povolené hodnoty položky <typ> v media-level sekcii vo formáte protokolu SDP.

<typ>	Význam	*
m	Názov média a adresa pre transport.	Nie
i	Titulok média.	Áno
c	Informácie o spojení.	Áno
a	Atribúty média.	Áno

SDP slúžia riadky začínajúce refazcom 'a='. Správa vo formáte SDP sa uvádza v tele správy protokolu SIP. Ukážka správy vo formáte SDP je nižšie:

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 10.47.16.5
s=SDP Seminar
c=IN IP4 224.2.17.12/127
a=recvonly
m=audio 49170 RTP/AVP 0
m=video 51372 RTP/AVP 99
a=rtpmap:99 h263-1998/90000
```

Správa začína session-level sekciou. Prvý riadok obsahuje verziu protokolu SDP. Špecifikácia uvádza, že by sa mala používať hodnota "0". Nasleduje identifikátor sedenia, názov sedenia, informácie o spojení a session-level atribút s hodnotou *recvonly*, ktorý je platný pre všetky media-level sekcie. Za ním sú uvedené dve media-level sekcie, pričom jedna z nich má ešte ďalší, tentokrát media-level atribút.

3.2.2 Protokol RTP

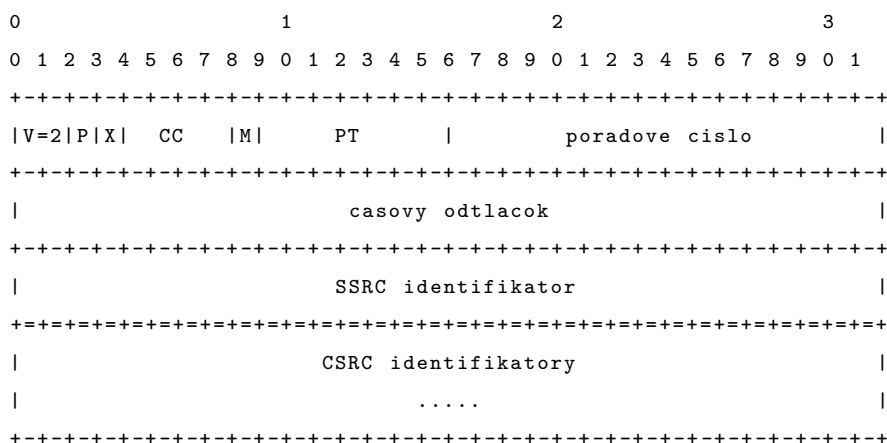
Real-time Transport Protocol (RTP) poskytuje služby prenosu multimediálnych údajov medzi dvoma a viacerými koncovými bodmi v paketovo orientovanej sieti. Tento protokol však negarantuje bezproblémové doručenie RTP paketov a neposkytuje ani kontrolu kvality služby. Je to rýdzo transportný protokol určený výhradne na prenos údajov, ktorého monitorovanie zabezpečuje protokol RTCP opísaný v ďalšej stati. Protokol je navrhnutý s dôrazom na to, aby mohol byť použitý v kombinácii s ľubovoľnou transportnou (napr. UDP, TCP) a sieťovou (napr. IPv4, IPv6) vrstvou modelu OSI [30].

Aj keď bol protokol pôvodne navrhnutý pre použitie v multimediálnych telekonferenciách s viacerými účastníkmi súčasne, nie je limitovaný len na tento prípad použitia. Prenos spojitých údajov a interaktívne distribuované simulácie sú len dva konkrétne prípady použitia z mnohých ďalších, ktoré protokol svojim dizajnom umožňuje. Najaktuálnejšia verzia protokolu je opísaná v IETF štandarde RFC 3550 [20].

Keďže je RTP binárne orientovaný protokol aplikačnej vrstvy, má význam spomenúť poradie bajtov a iné základné informácie. Všetky celočíselné hodnoty sa prenášajú v sieťovom poradí bajtov (bajt je v tomto prípade uvažovaný ako osem bitov), to znamená najvýznamnejší bajt ide prvý. Toto poradie bajtov sa zvyčajne označuje aj ako veľký endián. Ak nie je spomenuté inak, všetky číselné konštanty sú uvedené v desiatkovej sústave.

Všetky údaje v hlavičke paketu sú zarovnané na ich prirodzenú dĺžku. Napríklad 16-bitové polia sú zarovnané na párnú odchýlku (z angl. offset) a 32-bitové polia sú zarovnané na odchýlku deliteľnú číslom 4. To znamená, že sa v rámci zarovnania musia používať prázdne bity, ktoré majú hodnotu "0". Tento spôsob sa nazýva padding. Absolútny dátum a čas sa v RTP paketoch uvádza vo formáte NTP, ktorý

je bližšie opísaný v oficiálnej špecifikácii protokolu RTP [20]. Formát hlavičky RTP paketu je uvedený nižšie. Na výpise je znázornená postupnosť na úrovni bitov.



Prvých dvanásť bajtov hlavičky je pevných a obsahuje ich každý RTP paket. Zvyšné bajty hlavičky, označené ako CSRC identifikátory sú voliteľné. Za nimi nasleduje voliteľné rozšírenie hlavičky RTP paketu v prípade, že bit X má hodnotu “1”. Formát rozšírenia hlavičky je uvedený ďalej v tejto stati. Paket končí samotnými multimedialnými údajmi (payload), ktorých formát je pre každý kodek špecifický a je vždy opísaný niektorým doplňujúcim IETF štandardom. Rozšírenie niektorého formátu pre posielanie multimedialných údajov o posielanie prídavných informácií by mohlo mať význam z hľadiska možnosti jednoduchšej synchronizácie snímok s údajmi. Tejto možnosti sa budem venovať v podkapitole 4.2. Jednotlivé polia RTP paketu majú nasledujúci význam:

V je verzia použitého RTP protokolu. V súčasnej špecifikácii má hodnotu “2”.

P je príznak, či paket na konci obsahuje padding bajty pre zarovnanie.

X je príznak, či paket obsahuje rozšírenie hlavičky RTP.

CC obsahuje počet položiek, označených ako CSRC identifikátory.

M je popisovací bit, ktorého význam je závislý na použitom RTP profile.

PT je typ prenášaných údajov v tele RTP paketu, ktoré sú vždy na úplnom konci paketu.

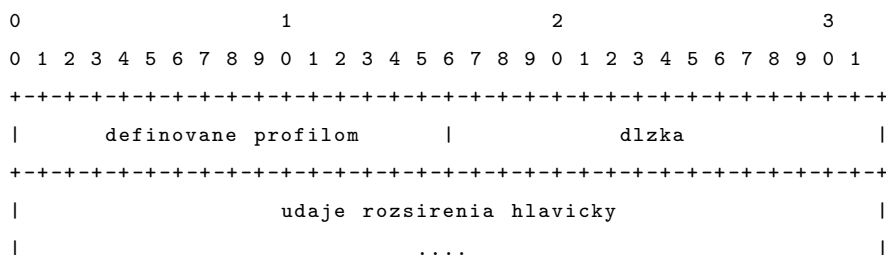
Poradové číslo označuje sekvenčné poradie odoslaného RTP paketu.

Časový odtlačok označuje časové poradie RTP paketu.

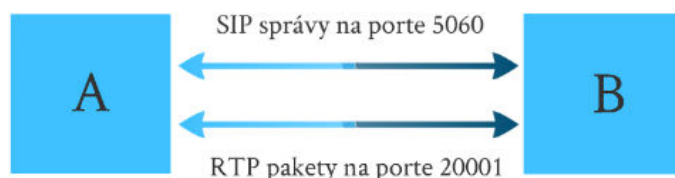
SSRC identifikátor je identifikátor synchronizačného zdroja. Synchronizačný zdroj jednoznačne identifikuje skupinu paketov z jedného zdroja. Napríklad výstup z mikrofónu a kamery sú rozdielne synchronizačné zdroje.

CSRC identifikátory je zoznam identifikátorov prispievajúceho zdroja. Zoznam prispievajúcich zdrojov jednoznačne identifikuje, z ktorých synchronizačných zdrojov je paket zložený. Používa sa napríklad v konferenčných hovoroch, kde môže rozprávať viac účastníkov súčasne a počas toho sa ich pakety mixujú. Pomocou tohto zoznamu je možné identifikovať, ktorý účastník aktuálne rozpráva.

Ak teda RTP paket obsahuje voliteľné rozšírenie hlavičky, musí nasledovať hneď za hlavičkou RTP paketu (to znamená za CSRC identifikátormi), pričom sa ale rozšírenie uvádza ešte pred telom RTP paketu. Rozšírenie hlavičky pozostáva z dvoch polí pevnej dĺžky 16 bitov. Prvé pole označené ako *“definovane profilom”* môže obsahovať ľubovoľnú 16-bitovú hodnotu na identifikáciu údajov v tele rozšírenia. Druhé pole označené ako *“dlzka”* musí obsahovať dĺžku údajov rozšírenia hlavičky uvedenú v 32-bitových slovách. Do tejto dĺžky sa prvé dve polia (*“definovane profilom”* a *“dlzka”*) do dĺžky nezapočítavajú, takže *“0”* je platná dĺžka. Rozšírenie končí samotnými údajmi, ktoré majú definovanú dĺžku. Rozšíreniami hlavičiek RTP sa ešte budem zaoberať v podkapitole 4.3. Pre lepšiu predstavu však uvádzam na tomto mieste aj formát rozšírenia hlavičky:



RTP pakety sa v zmysle VoIP služieb prostredníctvom protokolu SIP posielajú v samostatnom toku údajov na porte definovanom v tele SDP správy, ktorá bola doručená v tele SIP správy. Vzťah protokolov RTP a SIP počas prebiehajúceho sedenia s ich rozdielnymi tokmi údajov sú znázornené aj na obrázku 3–3.



Obr. 3–3 Vzťah protokolov SIP a RTP v prebiehajúcom sedení.

3.2.3 Protokol RTCP

Real-time Transport Control Protocol (RTCP) je rozšírením protokolu RTP a slúži na monitorovanie prebiehajúceho prenosu RTP paketov pre účely kontroly kvality spojenia a následné variabilné prispôsobovanie parametrov. Je definovaný v štandarde protokolu RTP [20], ktorý som spomínal v predchádzajúcej stati. Protokol RTCP ďalej rozširuje IETF štandard RFC 3611 [21].

Protokol RTCP je založený na periodickom posielaní kontrolných paketov všetkým účastníkom sedenia, použitím rovnakého distribučného mechanizmu, aký sa používa pri údajových RTP paketoch. Musí byť teda dostupné multiplexovanie údajových a kontrolných paketov, napríklad použitím rôznych UDP portov. Protokol vykonáva tri dôležité funkcie:

1. Primárnou úlohou je poskytovať spätnú väzbu na kvalitu distribúcie údajov. Toto je neoddeliteľnou súčasťou transportného protokolu RTP a súvisí to s funkciami kontroly vykonávania a kontroly zahĺtenia. Spätná väzba môže byť použitá napríklad pre adaptívne prispôsobovanie parametrov kodekov určujúcich výstupnú kvalitu obrazu a zvuku.
2. RTCP pakety obsahujú kanonický názov (CNAME) RTP zdrojov. Kanonický názov RTP zdroja poskytuje teda informáciu o identite a prítomnosti jednotlivých účastníkov sedenia. Keďže sa SSRC identifikátor RTP paketu môže pri zistenom konflikte zmeniť, nedá sa s určitosťou pomocou SSRC zistiť prítomnosť a identita účastníkov.
3. Prvé dve funkcie vyžadujú, aby každý účastník odosielať RTCP pakety všetkým ostatným účastníkom sedenia. Tým pádom má každý účastník informáciu o počte a identite prítomných účastníkov. Toto číslo sa používa na vypočítanie pomeru, v akom sú posielané RTCP pakety voči RTP paketom, aby nedochádzalo k nadmernému zahĺteniu siete aj v prípade väčšieho množstva účastníkov sedenia. O výpočte tohto pomeru je napísaná v špecifikácii celá kapitola, avšak vo všeobecnosti sa odporúča, aby RTCP nezaberalo viac, ako 5% šírky pásma pozostávajúceho z RTP a RTCP paketov.

Špecifikáciu protokolu RTCP definuje niekoľko typov kontrolných informácií, ktoré nesú RTCP pakety rôzneho typu. Ide konkrétne o:

Hlásenia odosielateľa (SR), ktoré nesú štatistické informácie o odosielaných a prijímaných údajoch od účastníkov, ktorí sú aktívnymi odosielateľmi.

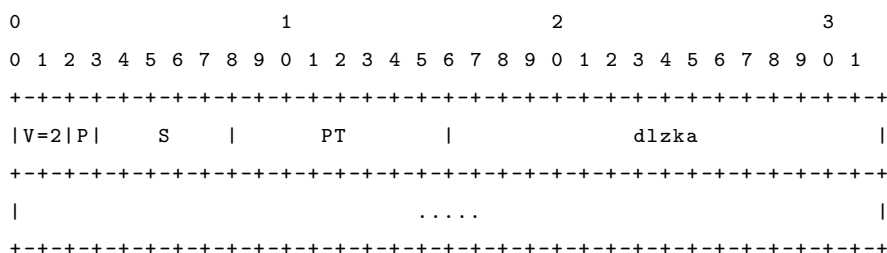
Hlásenia príjemcu (RR), ktoré nesú štatistické informácie o prijímaných údajoch od účastníkov, ktorí nie sú aktívnymi odosielateľmi.

Položky opisujúce zdroj (SDS), ktoré obsahujú napríklad spomínaný kanonický názov (CNAME).

Informácia o ukončení podielania sa na sedení (BYE).

Informácie špecifické pre konkrétnu aplikáciu (APP).

Každý RTCP paket začína pevnou časťou podobnou tej v RTP pakete, za ktorou nasledujú štruktúrované elementy variabilnej dĺžky, ktoré ale musia byť zarovnané na 32-bitovú hranicu. RTCP pakety sa dajú reťaziť bez akéhokoľvek oddeľovača, čím vznikne zložený RTCP paket, ktorý sa môže odoslať ako jediný paket protokolu nižšej vrstvy OSI modelu (napr. UDP). Z dôvodu rozsiahlosti uvádzam len formát pevnej časti hlavičky RTCP paketu. Každý z vyššie menovaných typov RTCP paketov má vlastný formát tela, ktorý je uvedený v špecifikácii [20] pre každý typ zvlášť.



Jednotlivé polia majú nasledujúci význam:

V je opäť ako v prípade RTP paketu verzia s rovnakou hodnotou, teda “2”.

P je príznak, či paket obsahuje na konci padding bajty pre zarovnanie.

S obsahuje hodnotu špecifickú pre každý typ RTCP paketu.

PT je typ RTCP paketu a jeho platné hodnoty sú: “200” (SR), “201” (RR), “202” (SDS), “203” (BYE) a “204” (APP).

Dĺžka obsahuje dĺžku RTCP paketu v 32-bitových slovách mínus jedna, takže “0” je platná dĺžka.

3.2.4 Ostatné protokoly

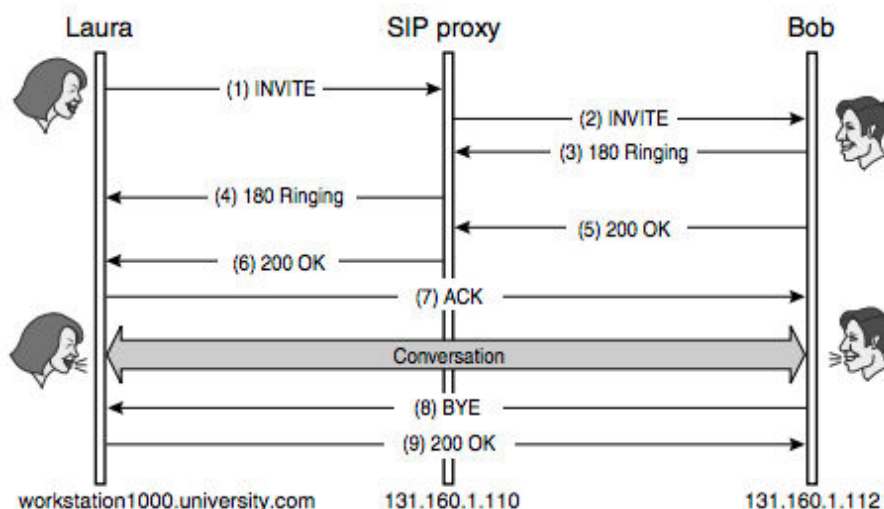
Funkcie protokolu SIP dopĺňajú okrem protokolov SDP, RTP a RTCP spomínaných v tejto práci aj ďalšie viac či menej dôležité štandardizované protokoly. V prvom rade ide napríklad o protokol XCAP [31], ktorý slúži na poskytovanie informácie o dostupnosti používateľov (online/offline stav). V rade druhom ide o protokol MSRP [32], ktorý dopĺňa podporu pre výmenu textových správ (instant messaging, IM). Z povahy tohto protokolu vyplýva, že je vhodný pre výmenu akýchkoľvek prídavných informácií a predstavuje teda jeden z troch spôsobov rozšírenia, ktorými sa budem zaoberať v kapitole 4. No a v neposlednom rade sú to protokoly STUN [25], TURN [26] a ICE [27], ktorých funkcia bude objasnená v podkapitole 3.4. Samozrejme, rozširujúcich a doplnkových protokolov, ktoré je možné aplikovať v kombinácii s niektorými preberanými protokolmi je oveľa viac. Sú však mimo rozsah tejto práce.

3.3 Ukážka komunikácie

Po tom, ako sme si objasnili, ako vyzerá základná štruktúra SIP správ, opísali vzťah protokolov SDP, RTP a RTCP s protokolom SIP a uviedli, aké typy logických entít sa zúčastňujú v SIP komunikácii je vhodné ukázať základný koncept fungovania na ucelenej ukážke, ktorá aj s obrázkami pochádza z [15]. Ukážka demonštruje nadviazanie VoIP komunikácie dvoch UA pomocou protokolu SIP za použitia proxy servera. Obidve menované entity boli spomenuté v stati 3.1.1. Na nasledujúcich stranách sa môžu vyskytnúť aj pojmy, ktoré neboli v predchádzajúcom texte explicitne vysvetlené. Budem sa ich preto vždy snažiť objasniť priamo v texte.

Na obrázku 3–4 sa Laura snaží dovolať Bobovi na jeho verejnú adresu, ale Bob sa na tejto adrese nenachádza. Proxy server na adrese *company.com* preto smeruje hovor na Bobovu aktuálnu adresu, ktorá je *sip:Bob@131.160.1.112*.

Táto ukážka sa skladá z troch rôznych SIP transakcií. Ide konkrétne o transakcie INVITE, ACK a BYE zo state 3.1.5. Na nasledujúcich riadkoch bude opísaná postupnosť vykonávania krokov pri nadviazovaní SIP hovoru z obrázku 3–4, v rámci ktorej budú vysvetlené princípy fungovania rôznych SIP transakcií s konkrétnymi výpismi vymieňaných správ protokolu SIP.



Obr. 3–4 Ukážka nadviazania hovoru pomocou protokolu SIP za použitia proxy servera. [15]

Krok číslo 1 na obrázku 3–4 znázorňuje prvú INVITE požiadavku, ktorú posielala Laurin UA na Bobovu verejnú adresu, pričom tú umiestňuje do úvodného riadku požiadavky za názov transakcie a do hlavičky *To*. Pridáva zároveň hlavičku *Via*, v ktorej uvedie vlastnú, teda Laurinu adresu a telo požiadavky vo formáte protokolu SDP, ktoré predstavuje Laurou ponúkané možnosti multimedialného sedenia. Laura chce prijímať RTP pakety obsahujúce zvukové údaje vo formáte PCM na porte 20002. Požiadavka je následne odoslaná proxy serveru na adrese *company.com*, pretože doménová časť Bobovej adresy za znakom “@” obsahuje práve túto adresu. Výpis tejto požiadavky vyzerá takto:

```

INVITE sip:Bob.Johnson@company.com SIP/2.0
Via: SIP/2.0/UDP workstation1000.university.com:5060
From: Laura Brown <sip:Laura.Brown@university.com>

```

```
To: Bob Johnson <sip:Bob.Johnson@company.com>
Call-ID: 12345678@workstation1000.university.com
CSeq: 1 INVITE
Contact: Laura Brown <sip:Laura@workstation1000.university.com>
Content-Type: application/sdp
Content-Length: 154

v=0
o=Laura 2891234526 2891234526 IN IP4 workstation1000.university.com
s=Let us talk for a while
c=IN IP4 138.85.27.10
t=0 0
m=audio 20002 RTP/AVP 0
```

V kroku číslo 2 na obrázku 3–4 je znázornená INVITE požiadavka, ktorú odosiela SIP proxy server po prijatí Laurinej INVITE požiadavky. Proxy server prečíta adresu požiadavky uvedenú na úvodnom riadku a zaujíma ho predovšetkým časť pred symbolom “@”, ktorá obsahuje refazec *Bob.Johnson*. Za použitia lokalizačnej služby, do ktorej sa každý UA musí na samom začiatku zaregistrovať pomocou REGISTER transakcie, v ktorej ohlásí svoju súčasnú adresu, proxy server vie, že Bob je zastihnuteľný na adrese *sip:Bob@131.160.1.112*. Vytvára teda novú INVITE požiadavku, ktorá je takmer identická s požiadavkou prijatou od Laury, v ktorej zamení cieľovú adresu v úvodnom riadku za súčasnú adresu, na ktorej je Bob zastihnuteľný podľa lokalizačnej služby a pridá ďalšiu hlavičku *Via*, v ktorej oznámi, že požiadavka prešla cez proxy server. Telo požiadavky ostáva nezmenené. Zmenená požiadavka vyzerá takto:

```
INVITE sip:Bob@131.160.1.112 SIP/2.0
Via: SIP/2.0/UDP 131.160.1.110
Via: SIP/2.0/UDP workstation1000.university.com:5060
From: Laura Brown <sip:Laura.Brown@university.com>
To: Bob Johnson <sip:Bob.Johnson@company.com>
Call-ID: 12345678@workstation1000.university.com
CSeq: 1 INVITE
Contact: Laura Brown <sip:Laura@workstation1000.university.com>
Content-Type: application/sdp
```

Content-Length: 154

```
v=0
o=Laura 2891234526 2891234526 IN IP4 workstation1000.university.com
s=Let us talk for a while
c=IN IP4 138.85.27.10
t=0 0
m=audio 20002 RTP/AVP 0
```

Bobov UA musí v kroku 3 z obrázku 3–4 po prijatí INVITE požiadavky od SIP proxy servera zahájiť upozornenie (väčšinou vyzváňaním) na prichádzajúci hovor od Laury a SIP proxy serveru vráti dočasnú odpoveď na prijatú INVITE požiadavku. Do dočasnej odpovede sa skopírujú všetky *Via* hlavičky z došlej INVITE požiadavky. Tie zabezpečia, že dočasná odpoveď prejde najprv cez proxy server na adrese *131.160.1.110* a následne bude proxy serverom preposlaná Laurinmu UA na adrese *workstation1000.university.com* a UDP porte 5060. Do dočasnej odpovede sa ďalej pridá *Contact* hlavička, ktorá obsahuje SIP adresu, kde môže byť Bob zastihnuteľný priamo. Od tejto chvíle budú ďalšie Laurine požiadavky posielané priamo na Bobovu adresu. Okrem toho Bobov UA pridá do *To* hlavičky aj parameter *tag*, ktorý slúži na pomenovanie Bobovho UA. Parameter *tag* pomôže rozlíšiť viacero Bobových UA, ktoré sa proxy server pokúsi kontaktovať na rôznych adresách, čím je docielená možnosť byť prihlásený z viacerých adries súčasne. Dočasná odpoveď je znázornená nižšie:

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 131.160.1.110
Via: SIP/2.0/UDP workstation1000.university.com:5060
From: Laura Brown <sip:Laura.Brown@university.com>
To: Bob Johnson <sip:Bob.Johnson@company.com>;tag=314159
Call-ID: 12345678@workstation1000.university.com
CSeq: 1 INVITE
Contact: Bob Johnson <sip:Bob@131.160.1.112>
```

V ďalšom kroku s číslom 4 z obrázku 3–4 proxy server prijme dočasnú od-

poved' od Bobovho UA, z ktorej odstráni hlavičku *Via* obsahujúcu svoju vlastnú adresu a v tomto tvare prepošle celú dočasnú odpoveď na adresu, ktorá je uvedená v nasledujúcej *Via* hlavičke. Odpoveď je nižšie:

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP workstation1000.university.com:5060
From: Laura Brown <sip:Laura.Brown@university.com>
To: Bob Johnson <sip:Bob.Johnson@company.com>;tag=314159
Call-ID: 12345678@workstation1000.university.com
CSeq: 1 INVITE
Contact: Bob Johnson <sip:Bob@131.160.1.112>
```

Po tom, ako Bob zdvihne prichádzajúci hovor, jeho UA odošle Laure koncovú odpoveď (krok číslo 5 na obrázku 3–4), ktorá obsahuje opäť vo formáte protokolu SDP popis Bobom ponúkaných možností multimediálneho sedenia. Bob bude prijímať RTP pakety obsahujúce zvukové údaje na UDP porte 41000. Koncová odpoveď je uvedená vo výpise pod týmto odstavcom.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 131.160.1.110
Via: SIP/2.0/UDP workstation1000.university.com:5060
From: Laura Brown <sip:Laura.Brown@university.com>
To: Bob Johnson <sip:Bob.Johnson@company.com>;tag=314159
Call-ID: 12345678@workstation1000.university.com
CSeq: 1 INVITE
Contact: Bob Johnson <sip:Bob@131.160.1.112>
Content-Type: application/sdp
Content-Length: 154
```

```
v=0
o=Bob 2891234321 2891234321 IN IP4 131.160.1.112
s=Let us talk for a while
c=IN IP4 131.160.1.112
t=0 0
m=audio 41000 RTP/AVP 0
```

Proxy server po prijatí koncovej odpovede od Boba prepošle túto odpoveď rov-

nakým spôsobom (krok číslo 6 na obrázku 3–4), ako predtým preposlal dočasnú odpoveď. Inými slovami, najprv odstráni hlavičku *Via* so svojou vlastnou adresou a potom odošle výslednú koncovú odpoveď na adresu, ktorá je v nasledujúcej *Via* hlavičke. Ukážka zmenenej odpovede je nižšie.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP workstation1000.university.com:5060
From: Laura Brown <sip:Laura.Brown@university.com>
To: Bob Johnson <sip:Bob.Johnson@company.com>;tag=314159
Call-ID: 12345678@workstation1000.university.com
CSeq: 1 INVITE
Contact: Bob Johnson <sip:Bob@131.160.1.112>
Content-Type: application/sdp
Content-Length: 154

v=0
o=Bob 2891234321 2891234321 IN IP4 131.160.1.112
s=Let us talk for a while
c=IN IP4 131.160.1.112
t=0 0
m=audio 41000 RTP/AVP 0
```

Keď Laurin UA prijme koncovú odpoveď “200 OK”, odošle ACK požiadavku (krok číslo 7 na obrázku 3–4), tentokrát už priamo na adresu Bobovho UA, ktorého adresa je v *Contact* hlavičke nedávno prijatej odpovede. Týmto Laurin UA potvrdí, že dostal informáciu o prijatí hovoru od Bobovho UA. Výpis požiadavky je nižšie:

```
ACK sip:Bob@131.160.1.112 SIP/2.0
Via: SIP/2.0/UDP workstation1000.university.com:5060
From: Laura Brown <sip:Laura.Brown@university.com>
To: Bob Johnson <sip:Bob.Johnson@company.com>;tag=314159
Call-ID: 12345678@workstation1000.university.com
CSeq: 1 ACK
Contact: Laura Brown <sip:Laura@workstation1000.university.com>
```

Po obdržaní tejto požiadavky Bobovým UA sa nadviaže multimedialne sedenie na UDP portoch dohodnutých v predošlej komunikácii. Laurin UA odosiela svoj zvú-

kový záznam Bobovmu UA na UDP porte 41000 a Bobov UA odosiela svoj zvukový záznam Laurinmu UA na porte 20002. Zvukové údaje sú v niekoľko milisekundových intervaloch odosielané v tele RTP paketov, pričom šírka pásma sa automaticky prispôbuje na základe vymieňaných štatistík o kvalite hovoru v rámci RTCP paketov. Celé multimediálne sedenie už prebieha priamo, bez prítomnosti proxy servera v komunikácii. Nanešťastie však existujú aj výnimočné prípady, v ktorých nie je možné komunikovať priamo, opísané v podkapitole 3.4. V takýchto prípadoch sa používa na spoločnú komunikáciu RTP proxy, cez ktoré prechádzajú všetky RTP a RTCP pakety.

Akonáhle sa Bob rozhodne ukončiť hovor, jeho UA odošle BYE požiadavku priamo Laurinmu UA (krok číslo 8 na obrázku 3–4). Vo výpise nižšie si môžeme všimnúť, že celočíselná hodnota v *CSeq* hlavičke bola zvýšená o 1. *CSeq* hlavička nesie informáciu o sekvenčnom (resp. poradovom) čísle transakcie. BYE požiadavka patrí pod novú transakciu. Výpis odoslanej požiadavky je nižšie:

```
BYE sip:Laura@workstation1000.university.com SIP/2.0
Via: SIP/2.0/UDP workstation1000.university.com:5060
From: Laura Brown <sip:Laura.Brown@university.com>
To: Bob Johnson <sip:Bob.Johnson@company.com>;tag=314159
Call-ID: 12345678@workstation1000.university.com
CSeq: 2 BYE
Contact: Bob Johnson <sip:Bob@131.160.1.112>
```

Po prijatí Bobovej požiadavky o ukončenie hovoru Lauriným UA sa ukončí prebiehajúce multimediálne sedenie (v tomto prípade zvukové) a Laurin UA vráti Bobovmu UA odpoveď “200 OK” na jeho BYE požiadavku (krok číslo 9 na obrázku 3–4). Týmto je hovor považovaný za ukončený. Výpis poslednej odpovede je nižšie:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP workstation1000.university.com:5060
From: Laura Brown <sip:Laura.Brown@university.com>
To: Bob Johnson <sip:Bob.Johnson@company.com>;tag=314159
```

Call-ID: 12345678@workstation1000.university.com

CSeq: 2 BYE

Contact: Laura Brown <sip:Laura@workstation1000.university.com>

Predchádzajúca ukážka nadviazania VoIP hovoru prostredníctvom protokolu SIP demonštrovala len veľmi základné možnosti použitia. Ukážka neobsahovala overovanie identity používateľov, registráciu v lokalizačnej službe, šifrovanie komunikácie, obchádzanie prekladačov adres, ani mnohé ďalšie praktiky, o ktoré bol protokol SIP rozšírený inými IETF štandardmi, ktoré slúžia na rôzne účely a činia tak SIP veľmi komplexným, ale na druhej strane robustným protokolom.

3.4 Problémy protokolu SIP

Samozrejme, ako všetko ostatné, aj protokol SIP má svoje obmedzenia a slabiny. Množstvo z nich zhrňa Henrik Ingo vo svojom článku [12]. Pre úplnosť ich uvádzam v slovenskom jazyku aj vo svojej práci doplnené o vlastné postrehy.

3.4.1 Firewall

Úlohou firewallu v počítačovom systéme (PS) je kontrolovať prichádzajúce a odchádzajúce sieťové spojenia analyzovaním zachytených paketov a vyhodnotením rizík na základe množiny pravidiel. Týmto spôsobom zabezpečujú PS voči rizikám z vonkajšej siete a voči vynášaniu informácií zo siete vnútornej.

Funkcie firewallu však často obmedzujú komunikáciu filtrovaním prichádzajúcich RTP paketov. Rôzne implementácie firewallu sa však môžu v reakciách na prichádzajúce pakety odlišovať. To znamená, že aj spôsobov na ich obchádzanie je množstvo a teda je ťažké vytvoriť univerzálne riešenie.

Často však firewall dovoľuje UDP paketom prichádzajúcim z istej IP adresy

dosiahnúť PS (RTP pakety sú zabalené v UDP paketoch), ak sa z IP adresy PS, ktorý je za firewallom, predtým odoslali UDP pakety na IP adresu, z ktorej pochádza prichádzajúca komunikácia. Niektoré počiatočné RTP pakety sa pri tomto procese samozrejme strácajú odfiltrovaním firewallom. Táto metóda obchádzania obmedzení firewallov sa nazýva *UDP hole punching* a je bližšie opísaná v [22].

V ostatných prípadoch, keď firewall blokuje komunikáciu za každých okolností a na nadviazanie RTP spojenia nefunguje ani metóda *UDP hole punching*, je potrebný centrálny RTP proxy server, cez ktorý potom prechádza celá RTP komunikácia.

3.4.2 NAT (Network Address Translation)

Prekladač sieťových adries (Network Address Translator alebo skrátene NAT) zohrával veľmi dlho dôležitú úlohu v predlžovaní životnosti protokolu IPv4 [23]. Bolo to pomerne jednoduché, no krátkodobé riešenie na rýchlo sa blížiaci problém s nedostatkom IPv4 adries do obdobia, kým by sa navrhla, vyladila a celoplošne nasadila nová náhrada za IPv4 adresy v podobe protokolu IPv6. NAT modifikuje IPv4 adresy pri prechode paketov smerovačmi sieťovej komunikácie a často tým ukrýva pôvodnú adresu a port odosielateľa vo vnútornej sieti a nahrádza ich verejnou adresou a portom zariadenia na ktorom prebieha preklad adries. Podobne ako tomu bolo pri firewallle, aj v tomto prípade existujú rôzne implementácie prekladu adries, ktoré sú uvedené v [24] spolu s tabuľkou dosiahnuteľnosti, ak sú dva koncové body prepojené zariadeniami so systémom NAT.

Ako môžeme vidieť vo výpisoch SIP správ (resp. SIP dialógov) z podkapitoly 3.3, niektoré správy obsahujú IP adresy na viacerých miestach. Keďže sa tieto IP adresy používajú na priamu komunikáciu s UA, komunikácia medzi dvoma koncovými bodmi zlyhá, ak je niektorý z koncových bodov pripojený do siete internet cez zariadenie s podporou prekladu IP adries. Toto je závažný problém a obmedzenie

protokolu SIP.

Na elimináciu problémov spojených so zavedením prekladačov IP adres do smerovačov sieťovej komunikácie bolo zavedených a štandardizovaných niekoľko techník. Medzi ne patria STUN [25], TURN [26] a ICE [27]. Základná myšlienka SIP komunikácie za prítomnosti zariadení s NAT je zistenie vlastnej verejnej IP adresy, ktorá sa v komunikácii následne používa namiesto súkromnej IP adresy. Túto úlohu môže vykonávať aj SIP proxy server, ktorý po prijatí SIP požiadavky od UA porovná adresu UA prijatú v hlavičkách požiadavky a adresu, z ktorej požiadavka skutočne prišla.

3.4.3 Šifrovanie

Obidva protokoly, SIP aj RTP, podporujú šifrované spojenie dvoch UA, avšak veľká väčšina implementácií buď šifrovanie nepodporuje vôbec alebo má chybnú podporu šifrovanej SIP a RTP komunikácie.

V prípade protokolu SIP existujú minimálne dva problémy týkajúce sa šifrovania. Prvým z nich je šifrovanie komunikácie medzi UA a SIP proxy serverom, prípadne medzi dvoma SIP proxy servermi (každý UA môže totiž používať iný SIP proxy server). Druhým je šifrovanie spojenia dvoch UA, kde komunikácia kvôli obmedzeniam spomenutým vyššie musí prechádzať cez proxy server. Prvý problém je riešiteľný použitím TLS (Transport Layer Security) podobne ako u protokolu HTTPS.

Druhý problém je však komplikovanejší, pretože časť informácií, ktoré SIP správy obsahujú, musí byť prístupná pre proxy server na účely ďalšieho nasmerovania komunikácie. Tieto informácie teda nemôžu byť z logických dôvodov zašifrované, aby k nim proxy server mohol pristupovať. Na vzájomnú autentifikáciu volaného a volajúceho by mohli byť použité kryptografické algoritmy. V tom prípade by však bola

potrebná Public Key infraštruktúra (PKI) na výmenu kľúčov medzi potenciálnymi používateľmi. Preto sa aj dnes len zriedka používa šifrovaná SIP komunikácia.

V prípade RTP existuje štandard SRTP [28], čo nie je nič iné, ako zabezpečený protokol RTP (z angl. Secure RTP). Aj keď pridanie podpory šifrovania do už tak zložitého protokolu nie je najľahšia úloha, implementácia SRTP by bola priamočiara, ale nie je tomu tak z jedného dôvodu. Existujú totiž rôzne prístupy k výmene šifrovacích kľúčov potrebných pre šifrovanú RTP komunikáciu a rôzne implementácie zatiaľ nie sú stopercentne kompatibilné.

Z týchto dôvodov sa aj napriek existencii techník pre šifrovanie SIP a RTP komunikácie zatiaľ nezačal tento typ komunikácie používať celoplošne.

3.4.4 Zložitosť

Ako som už na konci podkapitoly 3.3 spomínal, jedným z problémov protokolu SIP je jeho komplexnosť. Aj napriek zdanlivej jednoduchosti a priamočiarosti mechanizmu INVITE požiadaviek a ich odpovedí sú problémovou časťou implementácií práve SIP dialógy. Okrem štandardu RFC 3261, ktorý opisuje samotný protokol SIP existuje vyše tridsať ďalších IETF štandardov, ktoré rozširujú základný štandard RFC 3261. Navyše sa v praxi používa viacero SIP “dialektov”, takže aj dve rôzne implementácie toho istého štandardu protokolu SIP môžu mať problémy s kompatibilitou, čo je v praxi na základe mnou vykonaných testov v náväznosti na ďalšiu kapitolu bežná skutočnosť.

Jednou z príčin je snaha o pridanie podpory všetkých vlastností klasickej telefónnej siete do protokolu SIP. V skutočnosti je naozaj možné uskutočniť SIP hovor prostredníctvom multimedialnej brány v klasickej telefónnej sieti. Kvôli tomu obsahuje SIP podporu príkazov, ktoré by pri použití v IP sieti nemali význam. Napríklad pri snahe nadviazať prostredníctvom protokolu SIP spojenie s klasickou telefónnou

linkou môže dôjsť k prípadu, kedy ženský hlas ohlási: “Číslo, ktoré voláte, práve nie je v dosahu.”. Inými slovami chybové hlásenie. Keďže sa toto nepovažuje za úspešné nadviazanie hovoru, nemôže byť ani účtované. Musí byť teda v protokole jasne odlíšené od úspešne nadviazaného hovoru. V prípade protokolu SIP to teda znamená odpoveď “200 OK”. To ale predstavuje problém, pretože SIP UA nemôže začať prijímať RTP pakety predtým, než obdrží odpoveď “200 OK”, potvrdzujúcu úspešné nadviazanie hovoru, ktorá obsahuje informácie potrebné pre nadviazanie RTP komunikácie, takže UA nemá možnosť prijať a následne prehrať oznámenie o nedostupnosti používateľovi. Z týchto dôvodov existuje štandard RFC 3960 [29], ktorý definuje spôsoby zaobchádzania s týmito oznámeniami a iné situácie, známe pod označením skoré médiá (z angl. early media).

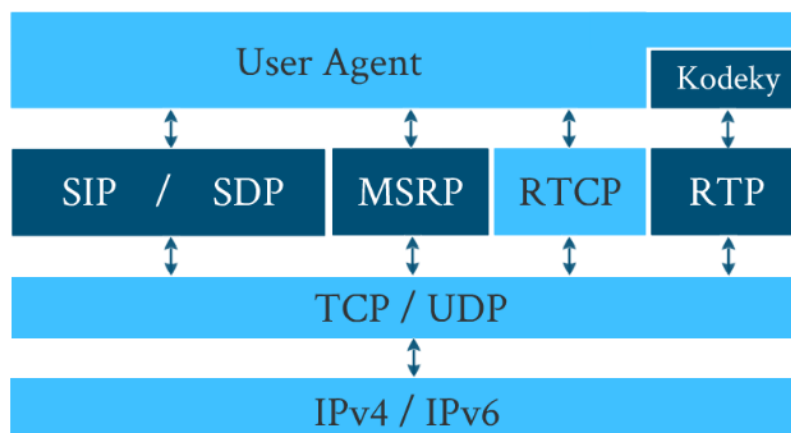
4 Návrh metódy prenosu údajov vzdialenej asistencie

Po relatívne stručnom úvode do komplexnej problematiky nadväzovania VoIP hovorov prostredníctvom protokolu SIP a opísaní niektorých možností protokolov rodiny SIP zhrnutých na obrázku 4–1 sa môžeme pustiť do návrhu spôsobu ich rozšírenia. Je potrebné hľadať optimálnu cestu prenosu prídavných informácií, ktoré by mohli byť prenášané a synchronizované s multimediálnym tokom obrazových, prípadne zvukových údajov. Požiadavky pre hľadanú metódu rozšírenia niektorého z protokolov rodiny SIP alebo ich kombinácie sú nasledovné:

- Generická implementácia, aplikovateľná pre rôzne video, audio, prípadne iné formáty údajového toku.
- Vybraný spôsob musí podliehať štandardom.
- Kompatibilita s existujúcou verejne prístupnou SIP infraštruktúrou.
- Možnosť synchronizácie prídavných informácií so vzorkami údajového toku (napr. snímky, úseky zvuku a pod.).
- Nízke nároky na objem prenášaných údajov.

Po lepšom zamyslení sa nad požiadavkami pre nový spôsob prenosu prídavných informácií a letmom pohľade na tmavou farbou vyznačené protokoly na obrázku 4–1 s prihliadnutím na informácie získané z kapitoly 3 sú hneď zrejmé niektoré možnosti rozšírenia.

Jednou z najjednoduchších možností je využiť na prenos týchto informácií flexibilný IM protokol MSRP, čomu je venovaná podkapitola 4.1. Tento protokol nebol v práci detailne opísaný z dôvodu rozsiahlosti a preto je potrebné sa počas analýzy držať jeho špecifikácie uvedenej v rámci štandardu RFC 4975 [32]. Ďalším potenciál-



Obr. 4 – 1 Protokoly rodiny SIP vo vzťahu k protokolom nižších vrstiev a k UA.

ným spôsobom rozšírenia by bolo posielat informácie priamo pribalené k obrazovým, prípadne zvukovým údajom do formátu pre kódovanie médií. O výhodách a nevýhodách tohto riešenia pojednáva podkapitola 4.2. Posledným spôsobom preberaným v tejto práci je využitie rozšírení hlavičky protokolu RTP, ktorým bude venovaná podkapitola 4.3.

Vzhľadom ku komplexnosti problematiky nadväzovania VoIP hovorov prostredníctvom protokolu SIP je vhodnejšie aplikáciu založiť na overenom aplikačnom rámci poskytujúcom príslušné možnosti rozšírenia, ktoré v rámci tejto práce hľadám. Preto je potrebné zároveň analyzovať existujúce implementácie protokolu SIP a vybrať vhodný aplikačný rámec pre dosiahnutie určeného cieľa.

4.1 Prenos pomocou protokolu MSRP

Ako prvé, pravdepodobne najjednoduchšie potenciálne riešenie prenosu prídavných informácií som skúmal použitie protokolu MSRP, pôvodne určeného pre instant messaging (IM) funkcionality. Za predpokladu použitia tohto spôsobu rozšírenia by síce nebolo možné synchronizovať prídavné informácie so snímkami priamo, čo znamená, že by boli prídavné informácie prenášané v rozdielnom toku údajov ako snímky, bolo

by však výrazne jednoduchšie ho implementovať, než nasledujúce dva spôsoby uvedené v ďalších podkapitolách. Jednoduchšie z toho pohľadu, že existujúce riešenia, ku ktorým sa dostaneme v závere celej analýzy a návrhu riešenia obsahujú podporu pre protokol MSRP a teda by stačilo definovať doménovo špecifický protokol pre prenos prídavných informácií.

Po zbežnom preštudovaní špecifikácie protokolu MSRP je vidieť jasné nevýhody použitia tohto spôsobu prenosu prídavných informácií. V ukážke použitia uvedenej pod týmto odstavcom je znázornený formát protokolu. Prvým problémom je to, že je príliš náročný na prenášaný objem údajov, keďže obsahuje množstvo redundantných položiek v hlavičkách, ktoré sú povinné a pre naše potreby nepoužiteľné. Navyše po každej odoslanej správe sa posiela späť aj potvrdenie o prijatí (potvrdenie začína na riadku “*MSRP a786hjs2 200 OK*”), čo znamená ďalšie redundantné údaje navyše, ktoré si v podmienkach mobilných sietí nemôžeme dovoliť. Podstatnú časť pritom tvorí len jediný riadok, na ktorom som pre názornú ukážku toho, čo je potrebné prenášať, uviedol zemepisnú šírku, dĺžku a smer natočenia.

```
MSRP a786hjs2 SEND
To-Path: msrp://biloxi.example.com:12763/kjhd37s2s20w2a;tcp
From-Path: msrp://atlanta.example.com:7654/jshA7weztas;tcp
Message-ID: 87652491
Byte-Range: 1-25/25
Content-Type: text/plain

48.730776 21.244640 68
-----a786hjs2$

MSRP a786hjs2 200 OK
To-Path: msrp://atlanta.example.com:7654/jshA7weztas;tcp
From-Path: msrp://biloxi.example.com:12763/kjhd37s2s20w2a;tcp
-----a786hjs2$
```

Je teda zrejmé, že namiesto 22 znakov prenášať rovných 400 nebude optimálnym riešením. Na druhej strane by bolo možné použiť iný, úspornejší typ obsahu

(hlavička *Content-Type*), ktorého formát by bol binárne orientovaný a dokázali by sme prídavné údaje prenášať efektívnejšie pomocou údajového typu s pohyblivou rádovou čiarkou v binárnej podobe pre údaje o GPS polohe (8 bajtov pre jeden double v jazyku Java), prípadne celočíselného údajového typu v binárnej podobe pre údaj o smere natočenia (2 bajty pre jeden short v jazyku Java). Ani to by však nevyriešilo prítomnosť pre naše účely nepodstatných hlavičiek a odpovede, ktoré predstavujú len zbytočnú redundanciu navyše.

4.2 Rozšírenia protokolu pre kódovanie multimedialných údajov

Kvôli vyššej efektivite prenosu a jednoduchšej synchronizácii prídavných informácií so snímkami som sa na základe odporúčaní členov CNL neskôr zamýšľal nad aplikáciou rozšírení na vrstve, v ktorej dochádza ku kódovaniu a dekódovaniu obrazových snímkov z rôznych video formátov, a ktorá je zobrazená na obrázku 4–1 ako “*Kodeky*”. Tieto formáty majú binárny charakter a mohli by mať výrazný dopad na zníženie množstva prenášaných údajov, ako som už naznačil v predchádzajúcej podkapitole.

Myšlienka v prvom rade vychádzala z flexibility formátu H.264, ktorý je o podobné meta údaje, respektíve špeciálne údaje používateľa rozšíriteľný pomocou SEI správ (z angl. Supplemental Enhancement Information). Ako už názov napovedá, ide o doplnujúce informácie, pripojiteľné k obrazovým snímkam v obálke formátu H.264. Syntax SEI správy je vyčerpávajúco opísaná v [33]. Konkrétnym typom SEI správy, ktorý by mohol slúžiť pre účely posielania prídavných informácií v toku obrazových údajov formátu H.264 je podľa oficiálneho štandardu ISO/IEC 14496-10:2012 [34] správa obsahujúca neregistrované používateľské údaje (z angl. user data unregistered).

Aj keď sú SEI správy pre formát H.264 špecifické, je táto myšlienka aplikovateľná aj vo všeobecnosti, keďže väčšina dnešných video formátov poskytuje rôzne spôsoby rozšírenia svojej obálky. Špecifickosť je však na druhej strane hlavná slabina tohto spôsobu rozšírenia, keďže sa vo VoIP službách často využíva veľká škála formátov a variabilná zmena ich parametrov sedenia z dôvodu kompatibility klientov. To by znamenalo potrebu implementácie rôznych spôsobov rozšírenia do viacerých typov video, prípadne audio dekodérov, čo by bolo málo efektívne z hľadiska neskoršej údržby. Preto bolo potrebné hľadať iné, generickejšie riešenie, ktorého implementácia by bola spoločná pre celú škálu formátov, nastavení a použitých nástrojov a tým pádom by bola z dlhodobého hľadiska ľahšie udržiateľná.

4.3 Rozšírenia hlavičky protokolu RTP

Je zrejmé, že predošlé dva spôsoby nesplňali požiadavky uvedené v kapitole 4. Hlavná nevýhoda použitia protokolu MSRP pre prenos prídavných informácií vzdialenej asistencie spočívala vo veľkom objeme prenášaných údajov a najväčším nedostatkom metódy využitia SEI správ formátu H.264 bola vysoká miera špecifickosti tohto mechanizmu. To znamená, že bolo potrebné hľadať vhodnú metódu rozšírenia na vyššej úrovni abstrakcie.

Túto úroveň abstrakcie predstavuje protokol RTP, ktorého pakety zodpovedajú za prenos zakódovaných multimediálnych údajov v ľubovoľnom podporovanom multimediálnom formáte (napr. aj spomínanom formáte H.264). Po zohľadnení informácií o protokole RTP získaných v stati 3.2.2 je hneď jasné, že tento protokol mechanizmus rozšírenia RTP paketov podporuje. Opisovaný mechanizmus má názov *“rozšírenia hlavičky protokolu RTP”*.

Podľa základnej špecifikácie protokolu RTP [20] však jeden RTP paket môže obsahovať nanajvýš jedno rozšírenie hlavičky protokolu RTP s maximálnou dĺžkou

65535 32-bitových slov, čo poskytuje dostatočný priestor pre všetky údaje posielané v multimediálnom toku údajov vo vzdialenej asistencii projektu Mapz. V rámci vzdialenej asistencie Mapz na druhej strane využívame minimálne 3 rôzne typy údajov, ktoré sa posielajú od nevidiaceho asistentovi. Aj napriek tomu, že by bolo možné posilať prídavné informácie jedného typu v jednom RTP pakete a tým pádom by pre nás stanovený limit dĺžky rozšírenia hlavičky RTP nezohrával dôležitú úlohu, bolo pre splnenie požiadavky maximálnej možnej efektivity prenosu údajov potrebné hľadať štandardizovaný spôsob, ktorý by umožnil do jedného RTP paketu vložiť viacero rôznych typov prídavných informácií.

Po preštudovaní stoviek strán rozširujúcich špecifikácii protokolu RTP a rôznych publikácií [35] o protokole RTP som tento spôsob našiel. RFC štandard s číslom 5285 [36] ho opisuje ako *“Všeobecný mechanizmus pre rozšírenia hlavičky RTP”*, ktorý bol navrhnutý na poskytnutie štandardizovaného mechanizmu, pomocou ktorého môžu byť v rámci rozšírenia hlavičky protokolu RTP posiellané rôzne typy údajov, ktoré majú jednoznačný identifikátor.

Pre správne použitie tohto mechanizmu rozšírenia podľa špecifikácie je potrebné vykonať dve hlavné úpravy protokolov SDP a RTP preberaných v podkapitole 3.2. Prvá úprava sa týka signalizácie multimediálnym sedením podporovaných rozšírení hlavičky RTP a využíva protokol SDP. Signalizácia podporovaných typov sa pre všetky prídavné informácie uskutočňuje prostredníctvom pridania riadku do tela protokolu SDP vo formáte:

```
a=extmap:<identifikator>/<smer> <uri>
```

Tento riadok sa pridáva buď do session-level sekcie alebo media-level sekcie. Pre jeden typ prídavných informácií sa však nesmie nachádzať v oboch sekciách. V jeho formáte sa predovšetkým uvádza jednoznačný identifikátor prídavnej informácie z platného rozsahu celočíselných hodnôt uvedeného ďalej v tejto kapitole

s výnimkou rezervovaných identifikátorov “0” a “15”. Ďalej sa uvádza smer, ktorý označuje, ktorým smerom je možné posielat, respektíve prijímať prídavné informácie. Povolené sú hodnoty:

“*inactive*” pre označenie, že tieto prídavné informácie daný klient nepodporuje v žiadnom smere.

“*sendonly*” pre označenie, že daný klient podporuje iba odosielanie týchto prídavných informácií.

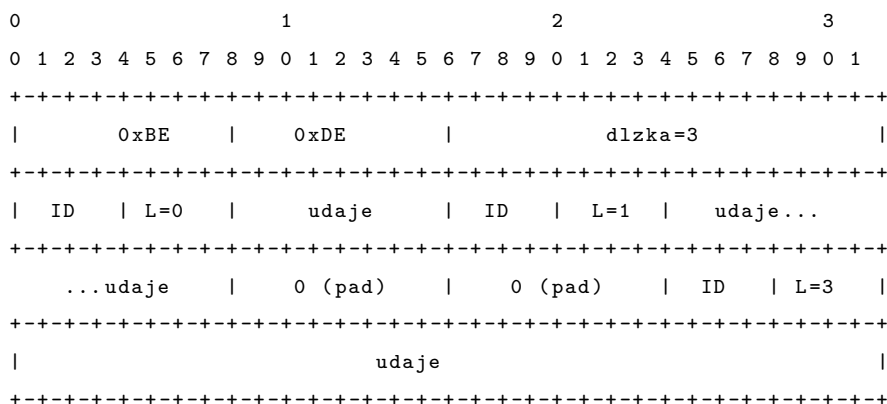
“*recvonly*” pre označenie, že daný klient podporuje iba prijímanie týchto prídavných informácií.

“*sendrecv*” pre označenie, že daný klient podporuje odosielanie a prijímanie týchto prídavných informácií.

Nakoniec sa uvádza URI identifikátor pre pomenovanie prídavných informácií v absolútnom tvare. Pre uvedenie konkrétneho príkladu použitia signalizačného mechanizmu v projekte Mapz uvádzam, ako by mohol vyzerat riadok pre signalizáciu podpory odosielania GPS polohy s identifikátorom “1” z aplikácie nevidiaceho:

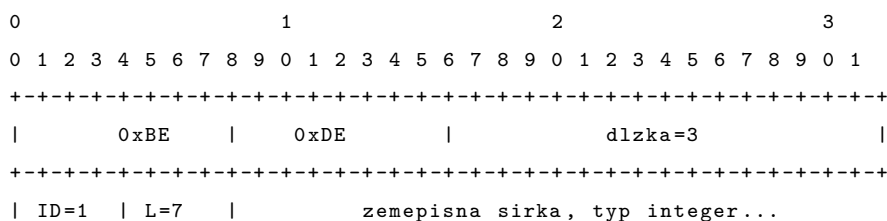
```
a=extmap:1/sendonly http://mapzproject.org/042013/ext.htm#gps-binary
```

Druhá úprava potrebná pre aplikáciu mechanizmu rozšírenia hlavičky protokolu RTP na základe štandardu RFC 5285 [36] sa týka samotného protokolu RTP. Konkrétne sa jedná o formát údajov hlavičky RTP, ktorý podľa špecifikácie môže byť uvedený v dvoch rôznych verziách - 1 alebo 2-bajtová verzia. Pre jednoduchosť tu opíšem len 1-bajtovú verziu. Celé rozšírenie hlavičky protokolu RTP vrátane základných dvoch polí označených ako “*definovane profilom*” a “*dlzka*” má pri použití opisovaného mechanizmu a 1-bajtovej verzie tvar:



V uvedenom formáte sú bajty *0xBE* a *0xDE* fixné a prostredníctvom týchto bajtov môže klient rozpoznať, v ktorej z dvoch verzií je uvedené telo rozšírenia hlavičky protokolu RTP. Bajty *0xBE* a *0xDE* sa používajú pre označenie 1-bajtovej verzie. Za týmito dvoma bajtmi sa uvádza dĺžka samotných údajov rozšírenia hlavičky v 32-bitových slovách. Ako údaje rozšírenia hlavičky sa uvádza množina elementov, kde každý element nesie jednoznačný identifikátor prídavnej informácie, dohodnutý počas signalizácie protokolom SDP. Za ním nasleduje dĺžka prídavnej informácie v bajtoch mínus jedna, takže hodnota "0" je povolená a znamená prítomnosť presne jedného bajtu údajov prídavnej informácie. Pre zarovnanie na celú dĺžku slova sa používajú takzvané padding bajty, ktoré sa môžu uvádzať za elementom alebo až na konci rozšírenia hlavičky.

Pre nadviazanie na predchádzajúcu demonštráciu použitia tohto mechanizmu v projekte Mapz uvádzam, ako bude vyzeráť samotné rozšírenie hlavičky, ktoré bude obsahovať prídavné informácie o GPS polohe nevidiaceho, ktorých signalizácia bola uvedená vyššie.



```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
...4 bajty |          zemepisna dlzka, typ integer...
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
...4 bajty |    0 (pad)    |    0 (pad)    |    0 (pad)    |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Rozšírenie teda začína bajtmi *0xBE* a *0xDE*, čo značí prítomnosť údajov rozšírenia hlavičky v 1-bajtovej verzii. Údaje rozšírenia hlavičky majú dĺžku troch 32-bitových slov. Telo obsahuje len jediný element, ktorého identifikátor je “1” (presne, ako bolo signalizované protokolom SDP vyššie) a dĺžka je 8 mínus jeden bajt, teda 7 bajtov. Nasledujú údaje elementu, ktorý nesie prídavné informácie o GPS polohe. Tie pozostávajú zo zemepisnej šírky a zemepisnej dĺžky, pričom obidve hodnoty sú po prenasobení ich pôvodnej hodnoty číslom 100000 typu integer, ktorý má v jazyku Java veľkosť 4 bajty. Telo je doplnené tromi padding bajtmi pre zarovnanie na dĺžku troch 32-bitových slov.

Prenos prídavných informácií o GPS polohe nevidiaceho využitím rozšírení hlavičky protokolu RTP si teda vyžaduje len 12 bajtov, čo je v porovnaní so súčasným riešením viac než dobré. To je možné dosiahnuť vďaka binárnej povahe protokolu RTP a veľmi jednoduchej kompresii prídavných informácií o GPS polohe, ktoré sú prenasobením ich pôvodnej hodnoty číslom 100000 (samozrejme, ak berieme do úvahy presnosť polohy na 5 desatinných miest, napr. 48.89364 a 2.33739) pretypované z typu double (8 bajtov) na typ integer (4 bajty). Toto riešenie splňa aj ostatné deklarované požiadavky a je teda preukázané, že z uvádzaných troch možných riešení prenosu prídavných informácií pre účely vzdialenej asistencie je toto riešenie najvhodnejšie. Preto sa bude implementácia zakladať práve na tejto metóde rozšírenia.

4.4 Analýza existujúcich implementácií

Na internete môžeme nájsť množstvo viac či menej spoľahlivých SIP implementácií vo forme serverových aplikácií, klientských aplikácií, separátnych RTP proxy serverov a aplikačných rámcov, ktoré sú zo zrejmých dôvodov pre účely tejto práce najpodstatnejšie. V nasledujúcich kapitolách len stručne uvediem zoznam implementácií, z ktorých budem skladať základnú platformu pre výsledný projekt. Pri výbere vhodných implementácií som kladol dôraz predovšetkým na to, aby mali otvorený zdrojový kód, ktorý je možné upraviť alebo rozšíriť.

4.4.1 Serverové aplikácie

Medzi zástupcami serverových aplikácií s podporou protokolu SIP a RTP môžeme nájsť rôzne riešenia s otvoreným zdrojovým kódom, ale aj rýdzo komerčné riešenia. Keďže je Mapz momentálne neziskový výskumný projekt, komerčné riešenia nás nezaujímajú. Medzi voľne dostupné alternatívy serverových implementácií, ktoré som mal možnosť vyskúšať vo virtuálnom prostredí a poskytovali značnú možnosť rozšírenia, patria:

- **Asterisk** je vyspelá VoIP platforma s podporou protokolu SIP. Asterisk je zároveň profesionálny produkt často nasadzovaný do komerčného prostredia. Počas jeho inštalácie mi spočiatku nebolo úplne jasné, akým spôsobom fungujú niektoré jeho časti, no napokon sa mi podarilo túto platformu vo virtuálnom prostredí dať do funkčného stavu. O staršej verzii Asterisk-u bolo napísaných niekoľko kníh, k tej najnovšej (v čase písania práce to bola verzia 11.3) však už toľko dostupných materiálov s výnimkou oficiálnej dokumentácie nebolo, čo značne sťažilo moje pokusy o konfiguráciu potrebnú pre účely vzdialenej asistencie projektu Mapz.
- **Kamailio** je ďalšia vyspelá VoIP platforma s podporou protokolu SIP. Ka-

mailio pôvodne vychádza z projektu OpenSER. Kamailio je veľmi ambiciózny projekt, o čom svedčí aj fakt, že som o tejto platforme našiel množstvo video návodov a nápomocných článkov od nezávislých blogerov. Po úvodnom zoznámení sa so spôsobom inštalácie a so spôsobom konfigurácie smerovania hovorov som si túto platformu obľúbil najviac zo všetkých mnou testovaných platformami. K tomu všetkému má Kamailio veľmi dobre spracovanú dokumentáciu. Platforma je kompatibilná s aplikáciou RTPproxy, ktorá, ako jej názov napovedá, slúži na prenos RTP paketov medzi klientmi, ktorí nemôžu komunikovať priamo. Kamailio je ľahko integrovateľný do plánovanej infraštruktúry vzdialenej asistencie projektu Mapz.

- **Freeswitch** je nádejný mladý projekt a škálovateľná VoIP platforma určená na spájanie známych komunikačných protokolov. Freeswitch poskytuje naozaj obrovské možnosti prispôbenia a ďalšieho rozšírenia. Konfiguračné súbory využívajú flexibilný formát XML. Jedným z problémov tejto platformy je v čase písania práce nevhodný spôsob ukladania a načítavania používateľských účtov do/zo súboru vo formáte XML. Na internete sa už vyskytli návody, ako dosiahnuť ukladanie a načítavanie používateľských účtov do/z MySQL databázy, tento spôsob sa mi však zdal byť prinajmenšom príliš ťažkopádny. Toto bol jeden z dôvodov, prečo som nakoniec zvolil platformu Kamailio. Existuje však aj možnosť použiť Freeswitch a Kamailio spoločne, kde Kamailio by bol zodpovedný za autentifikáciu používateľov, lokalizáciu používateľov, registráciu používateľov, smerovanie hovorov a informovanie o prítomnosti a Freeswitch by poskytoval iné služby, v ktorých vyniká.

4.4.2 Aplikačné rámce

Situácia okolo výberu aplikačných rámcov s podporou protokolu SIP a RTP na mobilných platformách bola najbiednejšia. Ukázalo sa, že v súčasnej dobe sú pre účely

projektu Mapz použiteľné pravdepodobne len dve implementácie, ktoré preukázali dostatočnú mieru kompatibility, vyspelosti a rozšíriteľnosti. Jedná sa o:

- **PJSIP** [42], ktorý vznikol ešte v roku 2005 a má za sebou dostatočne dlhú históriu. Je to vyspelý aplikačný rámec, ktorý obsahuje podporu pre väčšinu protokolov rodiny SIP. Je napísaný v jazyku C, jeho aplikačné rozhranie je podporované na rôznych operačných systémoch vrátane OS Android a v rôznych vyšších programovacích jazykoch. V rámci projektu PJSIP je distribuovaná aj ukážková aplikácia určená pre OS Android, ktorá bude spomenutá v ďalšej stati (CSipSimple). Tento aplikačný rámec obsahuje aj základnú podporu pre zvolenú metódu implementovaného rozšírenia uvedenú v podkapitole 4.3. Napriek tomu som si ho však nevybral ako východziu implementáciu kvôli absencii spoľahlivého a pre projekt Mapz kľúčového prenosu videa v jeho referenčnej implementácii určenej pre platformu Android (aplikácia CSipSimple).
- **doubango** [43] je relatívne mladý projekt, vytvorený spoločnosťou Doubango Telecom ako riešenie s otvoreným zdrojovým kódom, podobne ako PJSIP. Je tiež napísaný v jazyku C s bohatou podporou na rôznych platformách vrátane OS Android, iOS, Windows, Linux a iných. S aplikačným rámcom doubango je distribuovaná množina referenčných implementácií pre rôzne platformy, mimo iné aj OS Android (aplikácia IMSdroid spomínaná v ďalšej stati). Z pohľadu podpory protokolov rodiny SIP, podobne ako PJSIP obsahuje podporu pre väčšinu protokolov spomínaných v tejto práci. Zároveň obsahuje podobne ako PJSIP aj základnú podporu pre mnou navrhovanú metódu rozšírenia protokolov rodiny SIP. V implementačnej časti tejto práce teda opíšem rozšírenie tohto aplikačného rámca o podporu štandardu RFC 5285 vďaka jeho funkčnej a postačujúcej referenčnej implementácii pre platformu Android (aplikácia IMSdroid).

4.4.3 Klientské aplikácie

Z radu klientských aplikácií, učených pre mobilné platformy, s plnou podporou protokolu SIP a RTP na uskutočňovanie multimedialných hovorov (minimálne zvuk, obraz a text) a otvoreným zdrojovým kódom sa ukázalo, že väčšina má problémy s kompatibilitou nadviazania obrazového spojenia, ktoré je pre projekt Mapz kľúčové. Spomedzi mnou testovaných alternatív mobilných klientských aplikácií s podporou protokolu SIP uvádzam tie, u ktorých som videl najväčší potenciál budúceho rozšírenia:

- **CSipSimple** [40], ktorý mi bol pôvodne odporúčaný členmi CNL ako náhrada za aplikáciu SIPdroid, ktorej implementácia technológie SIP nie je na najlepšej úrovni. CSipSimple využíva existujúcu implementáciu technológie SIP s názvom PJSIP opísanú v stati 4.4.2. CSipSimple mal v čase písania práce podporu pre posielanie videa v skorej beta verzii, ktorú bolo potrebné doinštalovať dodatočným balíkom a napokon sa po testoch ukázala ako málo spoľahlivá. Obrazové spojenie sa často nepodarilo nadviazať alebo boli prenesené obrazové snímky poškodené.
- **IMSdroid** [41], ktorý som objavil počas hľadania alternatívy k aplikácii CSipSimple. Po vykonaní praktických testov sa ukázal byť ako vhodná východzia aplikácia pre ďalšie rozšírenia. IMSdroid taktiež využíva existujúcu implementáciu technológie SIP s názvom doubango, ktorá bola opísaná v stati 4.4.2. Podpora audio a video prenosu sa zdala byť dostatočne spoľahlivá pre účely projektu Mapz. Implementácia teda bude vychádzať z tejto aplikácie.

4.4.4 Ostatné aplikácie

K ostatným aplikáciám som zaradil len jednu mnou testovanú aplikáciu, ktorá predstavuje RTP proxy server s príznačným názvom RTPproxy. Nanešťastie som nemal

príležitosť overiť funkčnosť tejto aplikácie v reálnych podmienkach, kde firewall nepraje priamemu spojeniu a kde je prítomná zložitá sieťová štruktúra. Vo virtuálnom prostredí však fungovala bezchybne a je dokonca podporovaná a plne kompatibilná s niekoľkými serverovými aplikáciami.

5 Implementácia navrhovaného riešenia

V tejto kapitole bude opísaný postup implementácie mobilnej aplikácie založenej na platforme Android určenej pre účely vzdialenej asistencie v rámci projektu Mapz. Aplikácia sa bude zakladať na aplikačnom rámci doubango a jeho referenčnej implementácii IMSdroid, ktorá už obsahuje podporu pre vytváranie a prijímanie VoIP hovorov. Pre splnenie cieľov tejto diplomovej práce sa preto budem predovšetkým sústrediť na opis implementácie metódy prenosu prídavných informácií navrhovanej v podkapitole 4.3, do aplikačného rámca doubango. Funkčnosť tejto metódy bude demonštrovaná na vyvinutej mobilnej aplikácii založenej na referenčnej implementácii tohto aplikačného rámca (IMSdroid). Samotnej implementácii predchádzalo okrem pochopenia fungovania protokolov rodiny SIP aj štúdium fungovania a štruktúry aplikačného rámca doubango a spôsobu jeho integrácie s aplikáciou IMSdroid. Je dôležité poznamenať, že konfigurácia SIP servera a ani vývoj webovej aplikácie, o ktorých budem písať v podkapitole 5.1, nebudú opísané v tejto diplomovej práci.

Aby som teda zhrnul, implementáciou ktorých častí systému vzdialenej asistencie sa budem zaoberať v tejto práci, uvádzam, aké kroky je potrebné vykonať:

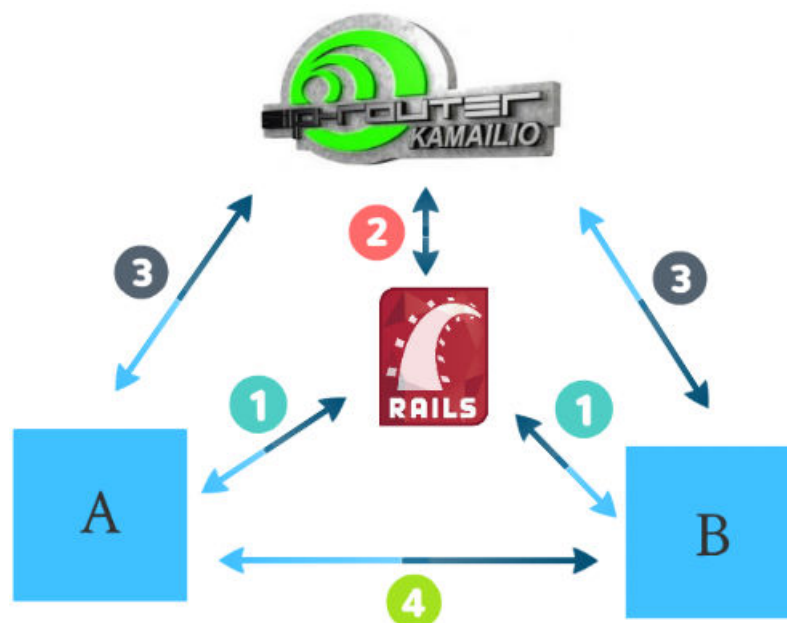
- Navrhnuť novú architektúru systému vzdialenej asistencie, ktorá bude využívať technológiu SIP na nadviazanie multimediálneho sedenia.
- Rozšíriť natívny aplikačný rámec doubango o metódu prenosu prídavných informácií navrhovaných v podkapitole 4.3.
- Upraviť SWIG obálku, ktorá sprístupní implementované mechanizmy na platforme Android.
- Upraviť aplikáciu IMSdroid, aby využívala implementované mechanizmy prostredníctvom SWIG obálky a splňala tým požiadavky pre systém vzdialenej asistencie.

5.1 Architektúra riešenia vzdialenej asistencie

Pre potreby implementácie navrhovaného riešenia bolo potrebné navrhnuť úplne novú architektúru, ktorá by dokázala pokryť požiadavky vzdialenej asistencie v projekte Mapz. Mnou navrhovaná architektúra celého riešenia je uvedená na obrázku 5–1.

Jednou zo súčastí celého navrhovaného riešenia je SIP server Kamailio, spomínaný v podkapitole 4.4, ktorý zodpovedá za registráciu, lokalizáciu, smerovanie a nadviazanie VoIP hovoru medzi asistentom a nevidiacim. Nakoľko bolo v našom prípade potrebné udržiavať v databáze používateľov dodatočné informácie o používateľskom účte (napr. typ používateľa - nevidiaci, asistent alebo administrátor) potrebné pre účely nášho projektu a nechcel som robiť priame zásahy do databázovej štruktúry servera Kamailio v záujme udržať server kompatibilný s budúcnymi aktualizáciami a záplatami, pridali sme do platformy vlastnú web aplikáciu založenú na webovom rámci Ruby on Rails pre umožnenie správy používateľských účtov v systéme Mapz. Webové služby, ktoré sú súčasťou tejto web aplikácie zároveň zodpovedajú za úvodnú autentifikáciu používateľov do systému Mapz. Návratovou hodnotou autentifikačnej metódy je pridelená štandardná SIP adresa potrebná pre registráciu v signalizačnej službe. Registráciou sa v tomto prípade myslí proces odoslania úvodnej REGISTER požiadavky na SIP server, ktorá poskytne SIP serveru aktuálnu IP adresu, na ktorej je používateľ zastihnuteľný v prípade pokusu o nadviazanie spojenia iným používateľom. Posledným prvkom architektúry riešenia je mobilná aplikácia založená na platforme Android (znázornená ako účastníci *A* a *B*), vychádzajúca z referenčnej implementácie vybraného aplikačného rámca, ktorá dokáže nadviazať hlasové a obrazové spojenie medzi nevidiacim a asistentom.

Detailnejší opis funkcie jednotlivých častí architektúry počas životného cyklu mobilnej aplikácie je na základe obrázku 5–1 nasledovný:



Obr. 5 – 1 Architektúra implementovaného riešenia vzdialenej asistencie Mapz.

1. Pri spustení mobilnej aplikácie sa používateľ prihlási so svojími prihlasovacími údajmi, ktoré obdržal od systému Mapz. Overenie správnosti údajov a získanie identity používateľa sa vykonáva odoslaním požiadavky na webové služby vytvorené v zmysle REST požiadaviek.
2. Webové služby majú priamy prístup do databázy SIP servera a získavajú potrebné informácie pre overenie identity používateľa. Klientovi vracajú informácie o účte a jemu pridelenej SIP adrese.
3. Po získaní identity používateľa a SIP adresy potrebnej pre registráciu v SIP službe, klient odosiela REGISTER požiadavku na SIP server. Týmto používateľ oznámi IP adresu, na ktorej bude odteraz zastihnuteľný v prípade pokusu o nadviazanie hovoru. Zároveň tak poskytne informáciu SIP serveru o svojej prítomnosti. Teraz sa už môže ktorýkoľvek používateľ pokúsiť o kontaktovanie iného používateľa, ktorého má v zozname kontaktov, tiež získanom prostredníctvom webových služieb.

4. Po nadviazaní VoIP hovoru prostredníctvom SIP servera v kroku číslo 3 sa vytvorí za ideálnych okolností priame P2P spojenie medzi účastníkmi a týmto sa začne prenos obrazových, zvukových a prídavných informácií medzi nevidiacim a asistentom.

5.2 Použité technológie

Z pohľadu použitých technológií počas implementácie vzdialenej asistencie založenej na VoIP službách bolo potrebné pochopiť a použiť hlavne tieto technológie:

- **SDK platformy Android** [45] - Základná sada štandardných a oficiálnych nástrojov a knižníc pre vývoj aplikácií na platforme Android od firmy Google. Na tejto platforme je založená aplikácia IMSdroid, z ktorej mobilná aplikácia Mapz vychádza.
- **NDK platformy Android** [46] - Rozšírená sada štandardných a oficiálnych nástrojov a knižníc pre vývoj kritických častí aplikácií v niektorom natívnom jazyku platformy C alebo C++. Týmto je umožnená efektívnejšia práca s pamäťou zariadenia a prístup k perifériám zariadenia na nižšej úrovni abstrakcie. Väčšina dnešných aplikácií založených na platforme Android nepotrebuje túto sadu nástrojov. Aplikačný rámec doubango je však z dôvodu prenositeľnosti medzi rôznymi operačnými systémami napísaný v jazyku C, takže jeho referenčná implementácia IMSdroid využíva NDK platformy Android kvôli potrebe integrácie s knižnicami aplikačného rámca doubango.
- **Android Annotations** [47] - Sada knižníc, ktorá uľahčuje vývoj Android aplikácií tým, že smeruje pozornosť programátora na implementáciu požadovanej logiky aplikácie namiesto rutinných častí zdrojového kódu, ktorý sa stará o úlohy, akými sú napríklad volanie časti kódu vo vlákne používateľského prostredia, naviazanie časti šablóny na členské premenné aktivity, získanie refe-

rencie na niektoré systémové služby alebo nastavenie udalosti, ktorá sa vykoná po kliknutí na niektoré tlačidlo. Okrem toho poskytuje podporu pre volanie REST webových služieb, ktoré naša architektúra obsahuje. Na volanie webových služieb využíva externú knižnicu Spring for Android [48] a na spracovanie odpovede z webových služieb sa využíva ľubovoľná externá knižnica, v našom prípade Jackson [49].

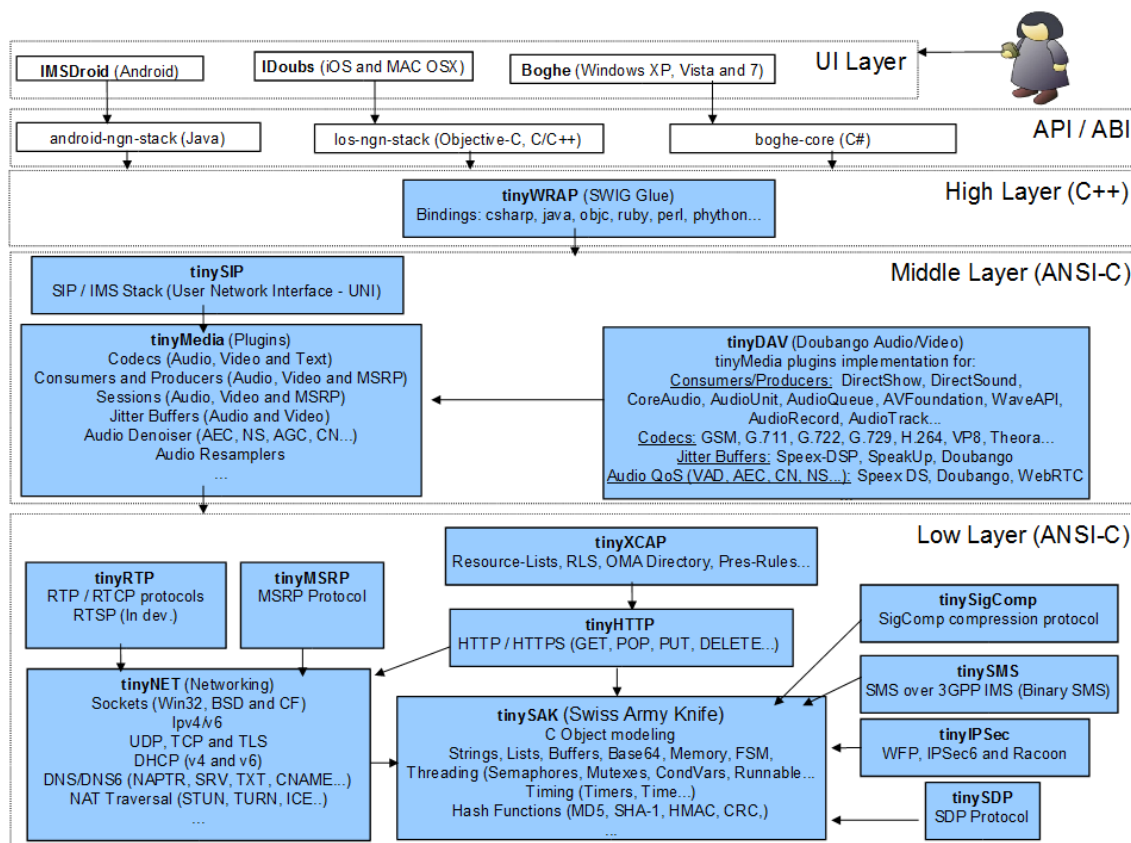
- **SWIG** [50] - Nástroj, ktorý spája aplikácie alebo knižnice napísané v programovacom jazyku C (resp. C++) s vyššími programovacími jazykmi, akými sú napríklad Perl, PHP, Python, Ruby, C#, Java a iné. SWIG sa využíva v aplikačnom rámci doubango na vygenerovanie obálky, ktorá prepája vrstvu napísanú v natívnom jazyku C s jazykom Java, používaným pre vývoj aplikácií na platforme Android.

5.3 Prenos prídavných informácií pre účely vzdialenej asistencie

Pre implementáciu navrhovanej metódy prenosu prídavných informácií o GPS polohe a smere natočenia nevidiaceho bolo na základe podkapitoly 4.3 potrebné rozšíriť zdrojový kód aplikačného rámca doubango, ktorého základná architektúra balíkov a ich náväznosti je uvedená na obrázku 5–2, o dva mechanizmy. Prvým z nich je mechanizmus signalizácie podporovaných rozšírení v jednotlivých údajových tokoch pomocou protokolu SDP. Druhým z nich je mechanizmus výmeny prídavných informácií v RTP paketoch na základe podmienok dohodnutých v SDP správe.

Počas implementácie som dbal predovšetkým na to, aby som nemenil aplikačné rozhranie rámca doubango spôsobom, ktorý by porušil kompatibilitu s budúcimi aktualizáciami a záplatami. To znamená, že som hľadal spôsoby, ako sa vyhnúť zmenám prototypov ktorejkoľvek vstavanej funkcie a namiesto toho som pridával

nové funkcie a nové členské premenné v štruktúrach pre sprístupnenie údajov na rôznych miestach v kóde, kde to bolo nevyhnutné pre splnenie cieľov tejto práce.



Obr. 5 – 2 Pohľad na balíky aplikačného rámca doubango. [43]

V nasledujúcich podkapitolách, ktoré pojednávajú o potrebných zmenách v jednotlivých balíkoch aplikačného rámca doubango sa budem odvolávať na balíky na základe ich názvu a preto uvádzam názvy a stručný opis balíkov, v ktorých som vykonal všetky zmeny potrebné pre dosiahnutie cieľa tejto práce:

tinyWRAP - V podstate ide o objektový model, respektíve vyššiu vrstvu abstrakcie natívneho aplikačného rámca. Tento balík obsahuje triedy napísané v jazyku C++ pre prácu s funkciami nižších vrstiev aplikačného rámca doubango. Tieto triedy sú priamo použiteľné v aplikáciách napísaných v jazyku C++ a po vygenerovaní SWIG obálky z týchto tried sú použiteľné aj vo vyšších

programovacích jazykoch.

tinyDAV - Funkcie pre prácu s multimediálnymi údajmi ako sú napríklad enkódovanie a dekodovanie týchto údajov s využitím rôznych externých knižníc kodekov.

tinyMEDIA - Funkcie pre prácu s multimediálnym sedením, nie však priamo s multimediálnymi údajmi.

tinyNET - Najnižšia komunikačná vrstva aplikačného rámca obsahujúca funkcie pre soketovú komunikáciu.

tinyRTP - Obsahuje funkcie určené pre spracovanie RTP paketov a RTCP paketov.

tinySAK - Obsahuje pomocné funkcie pre účely sledovania a odstraňovania chýb, prácu s pamäťou, prácu s vláknami a iné.

tinySIP - Obsahuje funkcie pre spracovanie SIP správ.

5.4 Rozšírenia aplikačného rámca doubango

V tejto podkapitole stručne opíšem, akým spôsobom som pridal podporu signalizácie podporovaných rozšírení hlavičky RTP a podporu pre posielanie rozšírení v RTP paketoch do aplikačného rámca doubango.

5.4.1 Signalizácia podporovaných rozšírení hlavičky RTP

Niektoré prídavné informácie vo vzdialenej asistencii projektu Mapz sú voliteľné. Ide napríklad o prídavné informácie zo senzorických zariadení voliteľného navigačného opasku. Okrem toho v dnešnej dobe existuje široká škála inteligentných mobilných

telefónov, ktoré sa líšia v ich senzorickom vybavení podľa triedy, do ktorej patria. Preto bolo potrebné na základe štandardu RFC 5285 implementovať aj mechanizmus signalizácie podporovaných rozšírení RTP, v tele ktorých budú podporované prídavné informácie prenášané.

Pre účely pridania podpory signalizácie podporovaných rozšírení hlavičky RTP bolo teda potrebné nájsť spôsob, pomocou ktorého by bolo možné z vyšších vrstiev (aplikácia, ktorá využíva aplikačný rámec *doubango*) nastaviť podporované rozšírenia daného sedenia. Na tento účel vhodne poslúžila trieda *ActionConfig* z balíka *tinyWRAP*, ktorá slúži na konfiguráciu rôznych nastavení, špecifických pre nadväzované VoIP sedenie. Do tejto triedy som pridal funkciu *addRtpHeaderExtensionSupport*, ktorá prijíma informácie potrebné pre signalizáciu podpory rozšírenia hlavičky RTP, opisované v podkapitole 4.3. Ide predovšetkým o ID rozšírenia, unikátnu URI adresu, smer prenosu a informáciu o tom, v rámci ktorých údajových tokov sa môže rozšírenie prenášať (audio, video, atď). Funkcia tieto informácie nastavuje do konfiguračného objektu sedenia nižšej vrstvy typu *tsip_action_t* (v balíku *tinySIP*), z ktorého sa počas vytvárania SIP/SDP správy vyčítavajú. Preto som vytvoril štruktúru *tmedia_rtp_header_extension_t*, ktorá obsahuje informácie o jednom podporovanom rozšírení hlavičky RTP. Konfiguračný objekt sedenia typu *tsip_action_t* teda obsahuje zoznam objektov tohto typu. Na základe týchto informácií sa potom do tela SDP správy pridáva riadok *a=extmap:<ID>/<smr> <URI>* pre každé podporované rozšírenie. Tento riadok sa pridáva vždy do príslušnej media-level sekcie SDP správy, podľa toho, v ktorých údajových tokoch bol povolený prenos prídavnej informácie.

Aby som umožnil na druhej strane aj vyčítavanie informácií o podporovaných rozšíreniach hlavičky RTP sedenia z tela prichádzajúcich SIP/SDP správ, bolo potrebné pridať do triedy *SipMessage* v balíku *tinyWRAP* funkciu *getSdpHeaderValueAtIndex*, ktorá vracia hodnoty riadkov začínajúcich na "a=" v SDP správe pre požadovanú media-level sekciu. Podporu pre analyzovanie prichádzajúcich SIP/SDP

správ a následné vyčítavanie konkrétnych hodnôt riadkov SDP správy pre jednotlivé sekcie na nižšej vrstve už aplikačný rámec doubango obsahoval.

5.4.2 Prenos dohodnutých rozšírení v RTP paketoch

Po vzájomnej dohode dvoch klientských mobilných aplikácií prostredníctvom signalizácie uvedenej v predchádzajúcej podkapitole je potrebné začať samotné prídavné informácie odosielať a na druhej strane prijímať. Bolo teda potrebné vytvoriť v balíku *tinyRTP* novú štruktúru s názvom *trtp_rtp_header_extension_t*, ktorá obsahuje údaje prídavnej informácie, ktorá je neskôr vkladaná do jedného elementu rozšírenia hlavičky protokolu RTP pod jedinečným ID v tvare opísanom v podkapitole 4.3.

Po bližšom skúmaní spôsobu, akým sa v pôvodnej implementácii aplikačného rámca doubango odosielajú multimediálne údaje som zistil, že ak by som chcel prídavné informácie pre účely ich synchronizácie priamo párovať s multimediálnymi údajmi, musel by som zmeniť prototyp funkcie *push* v triedach *ProxyAudioProducer* a *ProxyVideoProducer* z balíka *tinyWRAP* a prototyp callback funkcie typu *tmedia_producer_enc_cb_f* z balíka *tinyMEDIA*, čo by malo dopad na množstvo ďalších častí zdrojového kódu. Toto by okrem iných problémov znemožnilo splnenie stanovenej požiadavky o ponechaní možnosti kompatibility s budúcimi záplatami a aktualizáciami. Preto som sa rozhodol, že namiesto toho vytvorím v spomínaných triedach zásobník odchádzajúcich prídavných informácií, ktoré budú pridávané do najskorších odchádzajúcich RTP paketov a pridám metódu *addRtpHeaderExtensionData*, pomocou ktorej bude do zásobníka možné pridávať položky. Tým pádom nie je za každých okolností garantovaná synchronizácia prídavných informácií získaných v čase T so vzorkami údajového toku vytvorenými v rovnakom čase. Odchýlka v presnosti je však v tomto prípade zanedbateľná, keďže údaje odídu s najskorším odchádzajúcim RTP paketom. Zásobník odchádzajúcich prídavných informácií som pridal do štruktúry *trtp_manager_t*, ktorej objekty sú prístupné aj z balíka *tinyWRAP*, čo je vyššia

abstraktná vrstva, ktorá je už priamo využívaná mobilnou aplikáciou, aj z balíka *tinyRTP*, v ktorom je implementovaná serializácia a deserializácia RTP paketov. Serializáciu rozšírení hlavičky protokolu RTP do formátu opisovaného v podkapitole 4.3 bolo potrebné celú implementovať, keďže aplikačný rámec *doubango* podporu štandardu RFC 5285 neobsahoval. Na to slúžia funkcie v súbore *trtp_rtp_header_extensions.c*, ktoré sa nachádzajú v balíku *tinyRTP*.

Propagačný model prijatých prídavných informácií z nižších vrstiev aplikačného rámca (balíky *tinyNET* a *tinyRTP*) do vyšších vrstiev (balík *tinyWRAP*) sa zakladá na návrhovom vzore *Observer*. To znamená, že do tried *ProxyAudioConsumer* a *ProxyVideoConsumer* z balíka *tinyWRAP* bolo potrebné pridať funkciu *setRtpHeaderExtensionsCallback*, ktorá uloží referenciu na callback objekt typu *ProxyRtpHeaderExtensionsConsumerCallback* do statického zoznamu callback objektov, ktorý sa nachádza v balíku *tinyRTP* pod názvom *trtp_rtp_header_extensions_registered_cb_list*. Callback objekt obsahuje adresu funkcie, ktorú volá v prípade prijatia prídavných informácií v rozšírení hlavičky niektorého prichádzajúceho RTP paketu. Okrem toho obsahuje informácie o type multimediálneho toku údajov (audio alebo video), pre ktorý bol callback objekt zaregistrovaný. Po prijatí RTP paketu, ktorého hlavička má v bite *X* nastavenú hodnotu "1" sa spustí proces deserializácie rozšírení hlavičky RTP paketu podľa návrhu z podkapitoly 4.3, ktorý sa zakladal na RFC štandarde s číslom 5285. Funkcia pre deserializáciu rozšírení hlavičky RTP je v rovnakom súbore, v ktorom sa nachádza funkcia pre serializáciu. Po deserializácii jednotlivých prídavných informácií (každá v samostatnom elemente podľa návrhu) sa pre každú prídavnú informáciu volá funkcia typu *trtp_rtp_header_extensions_in_f* zo všetkých zaregistrovaných callback objektov.

5.5 Mobilná aplikácia pre vzdialenú asistenciu

Ako už bolo v tejto práci spomenuté, mobilná aplikácia, určená pre vzdialenú asistenciu vychádza z referenčnej implementácie aplikačného rámca doubango (IMSdroid). Implementovaná mobilná aplikácia dostala príznačný názov “*Vzdialený asistent*”. Táto aplikácia využíva aplikačný rámec doubango prostredníctvom obálky vygenerovanej z balíka *tinyWRAP* pomocou aplikácie SWIG. Táto obálka spolu s ďalším kódom, potrebným pre využívanie VoIP služieb sa nachádza v projekte *android-ngn-stack*. V tejto podkapitole opíšem, akým spôsobom funguje aplikácia *Vzdialený asistent* vo vzťahu k jej zdrojovým kódom.

5.5.1 Používateľské prostredie a spôsob fungovania

Aplikácia *Vzdialený asistent* funguje v dvoch režimoch podľa aktuálne prihláseného typu používateľa a svojím návrhom a rozmiestnením grafických komponentov je prispôbená pre použitie ako vidiacimi (asistenti), tak aj nevidiacimi používateľmi. Od verzie platformy Android 4.1.x totiž bola pridaná podpora pre ovládanie celého systému špeciálnymi dotykovými gestami, pričom sa používateľ presúva postupne po všetkých interaktívnych komponentoch (tlačidlá, textové polia a pod.) potiahnutím prsta naľavo alebo napravo na ľubovoľnom mieste displeja mobilného telefónu. Po prejdení na ľubovoľný interaktívny komponent je syntetizovaným zvukom oznámené označenie tohto komponentu. Potvrdenie sa aktivuje obyčajným dotykom na ľubovoľnom mieste displeja. Takýmto spôsobom je umožnená orientácia v celom operačnom systéme a ľubovoľnej aplikácii, na tento účel prispôbenej aj pre nevidiacich.

Spustením aplikácie sa vytvorí objekt triedy *VoipApplication* a spustí sa aktivita *LoginActivity*. Po prihlásení sa do aplikácie v aktivite *LoginActivity* sa používateľ ihneď dostane do zoznamu svojich kontaktov (aktivita *ContactsActivity*), z ktorého sa po otvorení detailov niektorého kontaktu (aktivita *ContactDetailActivity*) môže

pokúsiť nadviazať VoIP hovor s ľubovoľnou osobou zo zoznamu. Počas prihlasovania sa zároveň vytvorí objekt triedy *VoipEngine*, nad ktorým sa zavolá metóda *start* a týmto sa spustia všetky SIP služby z projektu *android-ngn-stack*. Počas spúšťania služieb sa spúšťa aj Android služba *VoipNativeService*, ktorá vytvorí objekt *BroadcastReceiver* a prihlási sa pomocou neho k odberu SIP udalostí, ktoré nastávajú v jednotlivých SIP službách počas komunikácie so SIP serverom ako napríklad oznámenie o prichádzajúcom hovore, oznámenie o prichádzajúcej odpovedi na odoslanú REGISTER požiadavku, informácia o prijatí hovoru volanou osobou a pod.

Po vytočení asistenta nevidiacim v aktivite *ContactDetailActivity* sa spustí aktivita *InCallActivity*. V rámci tejto aktivity je možné pomocou mechanizmu fragmentov platformy Android na základe aktuálneho stavu hovoru a typu používateľského účtu zobrazíť tri typy pohľadov:

- *BlindFragment*, ktorý sa používa počas prebiehajúceho hovoru na strane nevidiaceho.
- *AssistantFragment*, ktorý sa používa počas prebiehajúceho hovoru na strane asistenta.
- *CallRequestFragment*, ktorý sa používa počas čakania na prijatie hovoru volanou osobou a počas ukončenia hovoru.

5.5.2 Odosielanie a prijímanie prídavných informácií počas hovoru

Počas vytáčania sa na strane nevidiaceho zisťuje prítomnosť interného GPS zariadenia a interného digitálneho kompasu pre účely odosielania prídavných informácií. Informácia o dostupnosti týchto zariadení sa volaním metódy *addLocalRtpHeaderExtensionSupport* nad hlavným objektom multimediálneho sedenia typu *NgnAVSession* uloží. Toto spôsobí volanie funkcie *addRtpHeaderExtensionSupport* nad objektom typu *ActionConfig*, o ktorom som písal v stati 5.4.1. Mobilná aplikácia asistenta

po prijatí hovoru odosiela SIP odpoveď typu “200 OK” s údajom o podporovaných prídavných informáciách, ktoré je schopná spracovať. Týmto je zabezpečená kompatibilita medzi rôznymi verziami aplikácie *Vzdialený asistent* v prípade, že by sme v budúcnosti pridali podporu výmeny ďalších typov prídavných informácií. Mobilná aplikácia nevidiaceho po prijatí odpovede zlúči zoznam lokálne podporovaných prídavných informácií so zoznamom asistentom podporovaných prídavných informácií na základe špecifikácie uvedenej v štandarde RFC 5285. Toto sa udeje volaním funkcie *applyRtpHeaderExtensionsROtoLO* nad objektom multimediálneho sedenia typu *NgnAVSession*. Zlúčený zoznam podporovaných prídavných informácií sedenia potom odosiela v SIP správe typu *ACK*, potvrdzujúcej nadviazanie spojenia. V tejto chvíli aplikácia podporuje prídavné informácie týchto typov:

- **Údaje o GPS polohe**, ktoré majú pridelený identifikátor s číslom “1” a odosielajú sa v obrazovom toku údajov.
- **Údaje o smere natočenia nevidiaceho**, ktoré majú pridelený identifikátor “2” a odosielajú sa tiež v obrazovom toku údajov.

Po nadviazaní VoIP hovoru medzi nevidiacim a asistentom sa začne odosielať tok zvukových a obrazových údajov. Túto implementáciu som bez väčších úprav ponechal pôvodnú z aplikácie IMSdroid. Najväčšou úpravou bolo odstránenie možnosti nastavenia rôznych parametrov kodekov a ďalších podporovaných technológií. Keďže používatelia systému Mapz musia zo zrejmých dôvodov používať na komunikáciu len našu aplikáciu, nespôsobila táto zmena problémy s kompatibilitou, práve naopak. Keďže sa používa len jedna a tá istá aplikácia oboma typmi používateľov s rovnakými nastaveniami kodekov a ďalších parametrov, spojenie je za ideálnych podmienok väčšinou bezproblémové. Na enkódovanie obrazových údajov sa v tejto chvíli používa formát VP8 a na enkódovanie zvukových informácií sa používa formát G.722.

Počas prebiehajúceho hovoru sa v aplikácii nevidiaceho odoberajú aktualizácie GPS polohy z interného GPS zariadenia a aktualizácie smeru natočenia z interného digitálneho kompasu. Interval odoberania je v tejto chvíli nastavený na pevnú hodnotu tak, aby nedochádzalo k nadmernému zväčšovaniu objemu prenášaných údajov. Po získaní aktualizácie od niektorého zo spomínaných interných zariadení, volá sa nad objektom *NgnProxyVideoProducer* metóda *addRtpHeaderExtensionData* ktorá spôsobí pridanie prídavných informácií daného typu do zásobníka odchádzajúcich prídavných informácií na natívnej vrstve, keďže trieda *NgnProxyVideoProducer* dedí od triedy *ProxyVideoProducer*, upravenej pre tieto účely v stati 5.4.2. Prídavné informácie sa pri odoslaní najbližšieho RTP paketu s obrazovými údajmi vyberajú zo zásobníka a pridávajú ako rozšírenia hlavičky RTP. Jednotlivé typy údajov sa odosielať serializované v tomto tvare:

- **Údaje o GPS polohe** sa odosielať ako dvojica za sebou nasledujúcich celých čísel, každé typu *integer* s veľkosťou 4 bajty. Tieto celé čísla sa získavajú násobením pôvodných hodnôt typu *double* hodnotou 100000. Prvá v poradí je zemepisná šírka a za ňou nasleduje zemepisná dĺžka.
- **Údaje o smere natočenia nevidiaceho** sa odosielať ako typ *short* s veľkosťou 2 bajty. Vyjadruje sa v stupňoch ako uhol, pod ktorým je nevidiaci natočený relatívne od severu v smere hodinových ručičiek.

Na druhej strane ich asistent prijíma spôsobom, tiež opísaným v stati 5.4.2, pričom pri nadviazaní spojenia sa prihlásil k odberu týchto informácií registráciou callback objektu typu *NgnRtpHeaderExtensionsIncomingData* pomocou volania *setRtpHeaderExtensionsIncomingDataCallback* nad objektom typu *NgnProxyVideoConsumer*. Trieda *NgnProxyVideoConsumer* dedí od triedy *ProxyVideoConsumer*.

Po každom obdržaní prídavnej informácie v niektorom RTP pakete sa propaguje táto informácia z najnižších vrstiev balíkov *tinyNET* a *tinyRTP* cez balíky *tinyWRAP* a služby projektu *android-ngn-stack* až do fragmentu *AssistantFragment*

aplikácie *Vzdialený asistent*. V tomto fragmente sa prijatá informácia nakoniec de-serializuje a zobrazí v mape volaním metódy *setMarkerPosition* tohto fragmentu.

6 Vyhodnotenie použiteľnosti riešenia

Keďže aplikácia *Vzdialený asistent* je novšou verziou pôvodnej aplikácie určenej pre vzdialenú asistenciu v projekte Mapz, bolo potrebné určiť mieru zlepšenia oproti predchádzajúcemu stavu. V tejto kapitole na základe testov vykonaných v rôznych prostrediach určím kvalitatívne hodnoty vyvinutej aplikácie pri porovnaní s tou predchádzajúcou.

6.1 Metodológia testovania

Testy aplikácie *Vzdialený asistent* boli vykonané v rôznych prostrediach a pri rýchlo meniacej sa charakteristike kvality signálu v mobilnej sieti a Wi-Fi sieti. Testovanie prebiehalo na dvoch mobilných zariadeniach a taktiež medzi mobilným zariadením a osobným počítačom. Účastníci sa počas testovania nachádzali v rôznych lokalitách.

Pre účely merania doby oneskorenia prichádzajúcich obrazových údajov bolo potrebné zosynchronizovať čas na dvojici testovacích zariadení z rovnakého časového servera. Pre odmeranie doby oneskorenia bolo do zdrojového kódu potrebné pridať zaznamenávanie informácií o čase príchodu a odchodu paketov spolu s ich sekvenčnými číslami. Týmto spôsobom bolo možné porovnať časy príchodu a odchodu paketov a pripočítaním doby spracovania a zobrazenia snímok bolo možné určiť približnú hodnotu doby oneskorenia obrazu. Čas som meral s presnosťou na milisekundy.

Objem prenášaných obrazových údajov bol vypočítaný na základe znalosti formátov správ a na základe získaných informácií o veľkosti a tvare RTP paketov z programu Wireshark. Cieľom vyhodnotenia objemu prenášaných obrazových údajov nebolo určiť presný objem prenesených údajov za jednotku času, ale približne určiť, či sa objem oproti pôvodnej aplikácii zmenšil alebo zväčšil.

6.2 Vyhodnotenie doby oneskorenia obrazu

Na základe šiestich vykonaných meraní oneskorenia sa ukázalo, že počas hovoru v rámci Wi-Fi siete sa doba oneskorenia obrazu pohybovala od 150 do 200 milisekúnd pri maximálnej plynulosti obrazu bez vnímateľného trhania a výpadkov. Pre odtestovanie doby oneskorenia v mobilnej sieti bolo vykonaných ďalších šesť meraní, pričom sa ukázalo, že doba oneskorenia nesmierne kolíše a je závislá od aktuálneho vyťaženia siete v danej lokalite. Často sa stávalo, že pri pokuse o nadviazanie spojenia a zlej kvalite signálu mobilnej siete prišla úvodná INVITE požiadavka až po viac ako minúte. To mohlo byť spôsobené aj častým prepínaním technológie 3G a HSDPA. Priemerná hodnota doby oneskorenia po úspešnom nadviazaní spojenia sa pohybovala od 350 do 650 milisekúnd s vnímateľne nižším počtom zobrazených snímok za sekundu. Miestami obraz úplne vypadával, no po obnovení spojenia sa dal obraz automaticky znovu do pohybu.

V pôvodnej aplikácii vzdialenej asistencie Mapz sa doba odozvy pohybovala na úrovni od 800 do 1200 milisekúnd a mala stúpajúci charakter kvôli absencii mechanizmu kontroly kvality a variabilného upravovania šírky pásma. Okrem toho bol obraz viditeľne trhaný už aj vo Wi-Fi sieti. V tomto smere preto došlo k nesmiernemu zlepšeniu kvality.

6.3 Vyhodnotenie objemu prenášaných obrazových údajov

Ako som už uvádzal, pre porovnanie objemu prenášaných obrazových údajov bolo potrebné odchytať RTP pakety programom Wireshark. Na základe znalosti časového odtlačku (*Timestamp*) snímky z RTP paketu a informácie o úlohe *Marker* bitu v RTP pakete pre formát VP8 [51] som zistil, že jedna snímka vo formáte VP8 v aplikácii *Vzdialený asistent* je prenesená v 3 až 4 po sebe nasledujúcich RTP paketoch. Po odchytení dostatočného množstva RTP paketov, ktoré nesú obrazové

údaje bolo teda možné vypočítať veľkosť jednej snímky vrátane prídavných informácií a všetkých prenesených RTP hlavičiek. Priemerná veľkosť údajov prijatej jednej snímky s veľkosťou 355x288 pixelov sa pre formát VP8 pohybovala od 1900 bajtov do 2900 bajtov vrátane prídavných informácií a hlavičiek protokolu RTP všetkých 3, respektíve 4 paketov.

Ako som už v kapitole 1 naznačil, veľkosť jednej prenesenej snímky s potrebnými HTTP hlavičkami a prídavnými informáciami o GPS polohe a smere natočenia nevidiaceho bola v pôvodnej aplikácii vzdialenej asistencie projektu Mapz omnoho väčšia, ako vo vyvinutej aplikácii. Len údaje JPEG snímky s rovnakou veľkosťou 355x288 pixelov mali veľkosť 4000 až 6000 bajtov. Okrem týchto informácií bolo pre každú snímku potrebné prenášať aj textové údaje v kódovaní UTF-8, ktoré sú v porovnaní s binárnou povahou paketov súčasného riešenia neporovnateľne väčšie. Z toho môžeme vyvodiť tvrdenie, že aj v tomto smere vyvinutá aplikácia *Vzdialený asistent* v porovnaní s pôvodným riešením exceluje.

6.4 Celkové hodnotenie riešenia

Aj keď je na úplné dokončenie aplikácie *Vzdialený asistent* potrebné ešte nesmierné množstvo práce, implementovaná aplikácia je preukázateľne kvalitnejšia ako bola jej predošlá verzia v oboch opisovaných smeroch. To dokazujú nielen vyššie uvedené čísla, ale aj pocitové porovnanie kvalitatívnych parametrov aplikácie účastníkmi počas testovania. Autori kníh a iných odborných materiálov na tému použiteľnosti systémov reálneho času ([52] a [53]) určujú rozsah hodnôt doby oneskorenia, nad ktorými je už oneskorenie pocitovo vnímateľné na 150 - 200 ms. Riešenie *Vzdialený asistent* túto hranicu presahuje, no aj napriek tomu sa zdá byť na účely vzdialenej asistencie toto riešenie použiteľné. Príchod mobilných sietí LTE a nových, úspornejších kodekov akými sú VP9 (nástupca VP8) a HEVC (nástupca H.264) by mohol kvalitu tohto riešenia posunúť výrazne vpred.

7 Záver

Cieľom tejto diplomovej práce bolo nadviazať na predošlý výskum projektu Mapz v oblasti vzdialenej asistencie zrakovo postihnutých osôb a vyvinúť stabilnejšiu a kvalitnejšiu platformu pre nadväzovanie VoIP hovorov s hlavným zreteľom na navrhnutie nového spôsobu posielania prídavných informácií o GPS polohe a smere natočenia nevidiaceho v rámci multimedialného údajového toku pre potreby synchronizácie týchto informácií s obrazovými snímkami.

V práci boli analyzované tri možné spôsoby pre posielanie prídavných informácií, pričom len jeden z nich splňal určené kritéria. Tento spôsob sa zakladá na rozšíreniach hlavičky RTP paketu podľa štandardu RFC 5285. Pre overenie funkčnosti navrhovanej metódy bol rozšírený existujúci aplikačný rámec doubango o signalizáciu podporovaných rozšírení hlavičky RTP a posielanie prídavných informácií v rozšíreniach hlavičky RTP. Na demonštráciu použiteľnosti navrhovaného spôsobu v reálnej situácii pre účely vzdialenej asistencie bola vyvinutá mobilná aplikácia *Vzdialený asistent*, ktorá je založená na platforme Android.

Po vykonaných praktických testoch a meraniach kvalitatívnych parametrov, ktoré boli v pôvodnej aplikácii vzdialenej asistencie Mapz na výrazne nízkej úrovni sa ukázalo, že vyvinutá aplikácia *Vzdialený asistent* založená na technológii SIP naplnila všetky kvalitatívne očakávania. Napriek tomu je však aplikácia vo veľmi rannej fáze vývoja a k dokončeniu verejne prístupnej verzie je potrebné vykonať ešte množstvo práce a testov s cieľovou skupinou.

Literatúra

- [1] BUJACZ, M. - BARANSKI, P. - MORANSKI, M. - STRUMILLO, P. - MATERKA, A.: *Remote Guidance for the Blind - A Proposed Teleassistance System and Navigation Trials* [online]. Technical University of Lodz, Lodz, Poland, 2008 [cit. 2013-03-29]. Dostupné na internete: http://www.eletel.p.lodz.pl/programy/naviton/index.php?option=com_docman&task=doc_download&gid=199&Itemid=43.
- [2] BARANSKI, P. - STRUMILLO, P. - BUJACZ, M. - MATERKA, A.: *A Remote Guidance System Aiding the Blind in Urban Travel* [online]. Institute of Electronics, Technical University of Lodz, Lodz, Poland, 2009 [cit. 2013-03-29]. Dostupné na internete: http://www.eletel.p.lodz.pl/programy/naviton/index.php?option=com_docman&task=doc_download&gid=228&Itemid=43.
- [3] BARANSKI, P. - POLANCZYK, M. - STRUMILLO, P.: *A Remote Guidance System for the Blind*. IEEE International Conference on e-Health Networking, Lyon, France, 2010-07-03, s. 386-390. ISBN 978-1-4244-6375-6.
- [4] KOLEY, S. - MISHRA, R.: *Voice Operated Outdoor Navigation System for Visually Impaired Persons*. International Journal of Engineering Trends and Technology, 2012, vol. 3, no. 2, s. 153-157. ISSN 2231-5381.
- [5] AL-SALIHI, N.: *Precise Positioning in Real-Time for Visually Impaired People using Navigation Satellites*. International Journal of Engineering & Technology, 2012, vol. 12, no. 2, s. 83-89. ISSN 2077-1185.
- [6] GARAJ, V.: *Using Remote Vision to Navigate Visually Impaired Pedestrians: The Effects of Video Image Resolution on the Performance* [online]. School of Engineering and Design, Brunel University, United Kingdom [cit. 2013-

- 03-29]. Dostupné na internete: <http://www.iiis.org/cds2008/cd2008sci/CITSA2008/PapersPdf/I677HK.pdf>.
- [7] PORUBĀN, Jaroslav - PRISTÁŠ, Michal - STANÍK, Tomáš - ŠUŠČÁK, Marek - TOMÁŠ, Marek: *MapzProject.org — Main Page* [online]. KPI, FEI, Technická univerzita v Košiciach, 2011 [cit. 2013-03-29]. Dostupné na internete: <http://www.mapzproject.org/>.
- [8] TOMÁŠ, M.: *Platforma pre kolaboratívne získavanie a prezentáciu kontextovo závislých informácií so zreteľom na nevidiacich*: Bakalárska práca. KPI, FEI, Technická univerzita v Košiciach, 2011. s. 39-47.
- [9] ŠUŠČÁK, M.: *Využitie platformy Android pre získavanie, integráciu a prezentáciu priestorových informácií o službách*: Bakalárska práca. KPI, FEI, Technická univerzita v Košiciach, 2011. s. 45-49.
- [10] PRISTÁŠ, M.: *Využitie mobilných zariadení s Windows Phone 7 Application Platform*: Bakalárska práca. KPI, FEI, Technická univerzita v Košiciach, 2011. s. 41-43.
- [11] PEEK, B.: *MJPEG Decoder* [online]. Channel 9, [cit. 2013-03-29]. Dostupné na internete: <http://channel9.msdn.com/coding4fun/articles/MJPEG-Decoder>.
- [12] INGO, H.: *Session Initiation Protocol (SIP) and other Voice over IP protocols and applications* [online]. Sesca Technologies, Finland, [cit. 2013-03-29]. Dostupné na internete: <http://openlife.cc/system/files/FSWC%20Henrik%20Ingo%20Article%20SIP,%20VoIP%20and%20FLOSS.pdf>.
- [13] ROSENBERG, J. et. al.: *SIP: Session Initiation Protocol* [online]. IETF, RFC 3261, 2002. [cit. 2013-03-29]. Dostupné na internete: <http://tools.ietf.org/html/rfc3261>.

-
- [14] RADVISION Ltd.: *SIP: Protocol Overview*: White Paper [online]. 2001. Dostupné na internete: <http://www.radvision.com/nr/rdonlyres/51855e82-bd7c-4d9d-aa8a-e822e3f4a81f/0/radvisionsipprotocoloverview.pdf>.
- [15] GONZALLO, C.: *SIP Demystified*. McGraw-Hill, United States of America, 2002. ISBN 978-0071373401.
- [16] KLENSIN, J.: *Simple Mail Transfer Protocol* [online]. IETF, RFC 5321, 2008. [cit. 2013-03-29]. Dostupné na internete: <http://tools.ietf.org/html/rfc5321>.
- [17] FIELDING, R. et. al.: *Hypertext Transfer Protocol – HTTP/1.1* [online]. IETF, RFC 2616, 1999. [cit. 2013-03-29]. Dostupné na internete: <http://tools.ietf.org/html/rfc2616>.
- [18] HANDLEY, M. et. al.: *SDP: Session Description Protocol* [online]. IETF, RFC 4566, 2006. [cit. 2013-03-29]. Dostupné na internete: <http://tools.ietf.org/html/rfc4566>.
- [19] ROSENBERG, J. - SCHULZRINNE, H.: *An Offer/Answer Model with the Session Description Protocol (SDP)* [online]. IETF, RFC 3264, 2002. [cit. 2013-03-29]. Dostupné na internete: <http://tools.ietf.org/html/rfc3264>.
- [20] SCHULZRINNE, H. et. al.: *RTP: A Transport Protocol for Real-Time Applications* [online]. IETF, RFC 3550, 2003. [cit. 2013-03-29]. Dostupné na internete: <http://tools.ietf.org/html/rfc3550>.
- [21] FRIEDMAN, T. et. al.: *RTP Control Protocol Extended Reports (RTCP XR)* [online]. IETF, RFC 3611, 2003. [cit. 2013-03-29]. Dostupné na internete: <http://tools.ietf.org/html/rfc3611>.
- [22] FORD, B. - SRISURESH, P. - KEGEL, D.: *Peer-to-Peer Communication Across Network Address Translators* [online]. Massachusetts Institute of Tech-
-

- nology, Caymas Systems, Inc. Dostupné na internete: <https://pdos.csail.mit.edu/papers/p2pnat.pdf>.
- [23] KOZIEROK, M. C.: *TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference*. No Starch, 2005. s. 425. ISBN 978-1593270476.
- [24] WACKER, A. et. al.: *A NAT Traversal Mechanism for Peer-To-Peer Networks* [online]. University of Duisburg-Essen, Duisburg, Germany, University of Mannheim, Mannheim, Germany. Dostupné na internete: <http://www.pap.vs.uni-due.de/files/wacker-nat-traversal.pdf>.
- [25] ROSENBERG, J. et. al.: *Session Traversal Utilities for NAT (STUN)* [online]. IETF, RFC 5389, 2008. [cit. 2013-03-29]. Dostupné na internete: <http://tools.ietf.org/html/rfc5389>.
- [26] MAHY, R. et. al.: *Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)* [online]. IETF, RFC 5766, 2010. [cit. 2013-03-29]. Dostupné na internete: <http://tools.ietf.org/html/rfc5766>. ISSN: 2070-1721.
- [27] ROSENBERG, J.: *Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols* [online]. IETF, RFC 5245, 2010. [cit. 2013-03-29]. Dostupné na internete: <http://tools.ietf.org/html/rfc5245>. ISSN: 2070-1721.
- [28] BAUGHER, M. et. al.: *The Secure Real-time Transport Protocol (SRTP)* [online]. IETF, RFC 3711, 2004. [cit. 2013-03-29]. Dostupné na internete: <http://tools.ietf.org/html/rfc3711>.
- [29] CAMARILLO, G. - SCHULZRINNE, H.: *Early Media and Ringing Tone Generation in the Session Initiation Protocol (SIP)* [online]. IETF, RFC 3960,

2004. [cit. 2013-03-29]. Dostupné na internete: <http://tools.ietf.org/html/rfc3960>.
- [30] HILL, B.: *Cisco: The Complete Reference*. McGraw-Hill/Osborne Media, 2002. s. 3-19. ISBN 978-0072192803.
- [31] ROSENBERG, J.: *The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)* [online]. IETF, RFC 4825, 2007. [cit. 2013-03-29]. Dostupné na internete: <http://tools.ietf.org/html/rfc4825>.
- [32] CAMPBELL, B.: *The Message Session Relay Protocol (MSRP)* [online]. IETF, RFC 4975, 2007. [cit. 2013-03-29]. Dostupné na internete: <http://tools.ietf.org/html/rfc4975>.
- [33] ITU-T Recommendation H.264: Advanced Video Coding for Generic Audiovisual Services [online]. 2012-01. [cit. 2013-03-29]. s. 323. Dostupné na internete : http://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-201201-I!!PDF-E&type=items
- [34] ITU-T International Standard ISO/IEC 14496-10: Information technology - Coding of audio-visual objects. 2012-05-01, s. 346.
- [35] PERKINS, C.: *RTP: Audio and Video for the Internet*. Addison Wesley, 2003. ISBN 978-0672322495.
- [36] SINGER, D. - DESINENI, H.: *A General Mechanism for RTP Header Extensions* [online]. IETF, RFC 5285, 2008. [cit. 2013-03-29]. Dostupné na internete: <http://tools.ietf.org/html/rfc5285>.
- [37] ASTERISK: *Asterisk VoIP platform* [online]. [cit. 2013-03-29]. Dostupné na internete: <http://www.asterisk.org/>.

-
- [38] KAMAILIO: *Kamailio SIP server* [online]. [cit. 2013-03-29]. Dostupné na internete: <http://www.kamailio.org/>.
- [39] FREESWITCH: *Freeswitch VoIP platform* [online]. [cit. 2013-03-29]. Dostupné na internete: <http://www.freeswitch.org/>.
- [40] GOOGLE CODE: *CSipSimple - SIP application for Android devices* [online]. [cit. 2013-03-29]. Dostupné na internete: <https://code.google.com/p/csipsimple/>.
- [41] GOOGLE CODE: *imsdroid - High Quality Video SIP/IMS client for Google Android* [online]. [cit. 2013-03-29]. Dostupné na internete: <https://code.google.com/p/imsdroid/>.
- [42] PJSIP: *Open Source SIP, Media and NAT Traversal Library* [online]. [cit. 2013-03-29]. Dostupné na internete: <http://www.pjsip.org/>.
- [43] GOOGLE CODE: *doubango - cross-platform 3GPP IMS/LTE framework for embedded systems* [online]. [cit. 2013-03-29]. Dostupné na internete: <https://code.google.com/p/doubango/>.
- [44] RTPproxy: *RTP proxy server* [online]. [cit. 2013-03-29]. Dostupné na internete: <http://rtpproxy.org/>.
- [45] ANDROID DEVELOPERS: *Android SDK* [online]. [cit. 2013-03-29]. Dostupné na internete: <http://developer.android.com/sdk/index.html>.
- [46] ANDROID DEVELOPERS: *Android NDK* [online]. [cit. 2013-03-29]. Dostupné na internete: <http://developer.android.com/tools/sdk/ndk/index.html>.
- [47] EXCILYS GROUP: *Android Annotations* [online]. [cit. 2013-03-29]. Dostupné na internete: <http://androidannotations.org>.
-

- [48] SPRING SOURCE: *Spring for Android* [online]. [cit. 2013-03-29]. Dostupné na internete: <http://www.springsource.org/spring-android>.
- [49] CODEHAUS: *Jackson Java JSON-processor* [online]. [cit. 2013-03-29]. Dostupné na internete: <http://jackson.codehaus.org>.
- [50] SWIG: *Simplified Wrapper and Interface Generator* [online]. [cit. 2013-03-29]. Dostupné na internete: <http://www.swig.org/>.
- [51] WESTIN, P. et. al.: *RTP Payload Format for VP8 Video* [online]. IETF, draft-ietf-payload-vp8-08, 2013. [cit. 2013-03-29]. Dostupné na internete: <http://tools.ietf.org/html/draft-ietf-payload-vp8-08>.
- [52] SMARTVOX: *RTP, Jitter and audio quality in VoIP* [online]. 2012-04-24 [cit. 2013-03-29]. Dostupné na internete: <http://kb.smartvox.co.uk/voip-sip/rtp-jitter-audio-quality-voip/>.
- [53] NIELSEN, J.: *Usability Engineering*. Morgan Kaufmann, San Diego, NY, 1993. s. 135. ISBN 978-0125184069.
- [54] MILLER, L. - Gregory, P. H.: *SIP Communications for Dummies*. Wiley, 2009. ISBN 978-0-470-38114-4.

Zoznam príloh

Príloha A CD médium – diplomová práca v elektronickej podobe, prílohy v elektronickej podobe, zdrojové kódy riešenia.

Príloha B Používateľská príručka

Príloha C Systémová príručka

Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky

Rozšírenia SIP protokolu pre účely
vzdialenej asistencie v mobilných sieťach

Príloha B: Používateľská príručka

8 Používateľská príručka

Táto používateľská príručka obsahuje informácie, potrebné pre bezproblémovú inštaláciu a používanie mobilnej aplikácie *Vzdialený asistent*, vyvinutej pre účely tejto diplomovej práce. Aplikácia *Vzdialený asistent* je súčasťou komplexného systému Mapz vyvinutého v rámci výskumu týkajúceho sa vzdialenej asistencie zrakovo postihnutých osôb uskutočneného na pôde Technickej Univerzity v Košiciach v rokoch 2010, 2011 a 2012.

8.1 Funkcia aplikácie

Mobilná aplikácia *Vzdialený asistent* je určená pre použitie v rámci systému vzdialenej asistencie projektu Mapz a slúži na vytvorenie VoIP hovoru medzi dvoma účastníkmi, pričom jeden z nich je asistent a druhý zrakovo postihnutá osoba. Pomocou prenosu obrazových údajov, zvukových údajov a prídavných informácií prostredníctvom siete internet je asistentovi poskytnutý celkový pohľad na vzdialenú situáciu, v ktorej sa zrakovo postihnutá osoba nachádza. Prídavnými informáciami sa myslia údaje o GPS polohe a smere natočenia nevidiaceho. Týmto je asistentovi umožnené ľahšie sa v situácii zorientovať a poskytnúť nevidiacemu cenné informácie, ktoré mu kvôli jeho hendikepu nie sú známe. Asistent teda na diaľku čiastočne dopĺňa slabý alebo úplne chýbajúci zrakový vnem zrakovo postihnutej osoby.

8.2 Súpis obsahu dodávky

V zozname nižšie je uvedená hierarchia priečinkov na priloženom CD a zoznam súborov v každom z priečinkov. Súčasťou dodávky nie sú časti systému Mapz, ktoré neboli z pohľadu implementácie v práci detailne opísané (webové služby a nakonfigurovaný SIP server), pretože ich zdrojové kódy a konfiguračné súbory boli vyvinuté

ostatnými členmi výskumného projektu a teda podliehajú autorskému zákonu.

- *./src/*

RemoteAssistant/ - zdrojové kódy riešenia *Vzdialený asistent*.

- *./bin/*

RemoteAssistant.apk - inštalačný súbor aplikácie *Vzdialený asistent*

- *./doc/*

Generated/ - vygenerované súbory dokumentácie z programu *doxygen*

DiplomaThesis/ - zdrojové texty diplomovej práce vo formáte systému L^AT_EX

DiplomaThesis.pdf - diplomová práca vo formáte PDF

8.3 Hardvérové a softvérové požiadavky

Pred samotnou inštaláciou aplikácie *Vzdialený asistent* je potrebné sa ubezpečiť, že má používateľ k dispozícii vhodné mobilné zariadenie. Vhodným mobilným zariadením sa v tomto prípade myslí inteligentný mobilný telefón alebo tablet s operačným systémom Android vo verzii minimálne 4.2.2 a vyššie. Aplikácia *Vzdialený asistent* je totiž založená práve na tejto verzii platformy Android kvôli tomu, že táto verzia platformy obsahuje podporu pre prístupné ovládanie špeciálnymi gestami. Táto služba sa nazýva *TalkBack*. Používané mobilné zariadenie musí ďalej spĺňať tieto hardvérové požiadavky:

- 1200 Mhz procesor s minimálne dvoma jadrami,
- 1 GB pamäte RAM,
- Prítomnosť modulov GPS alebo GLONASS,

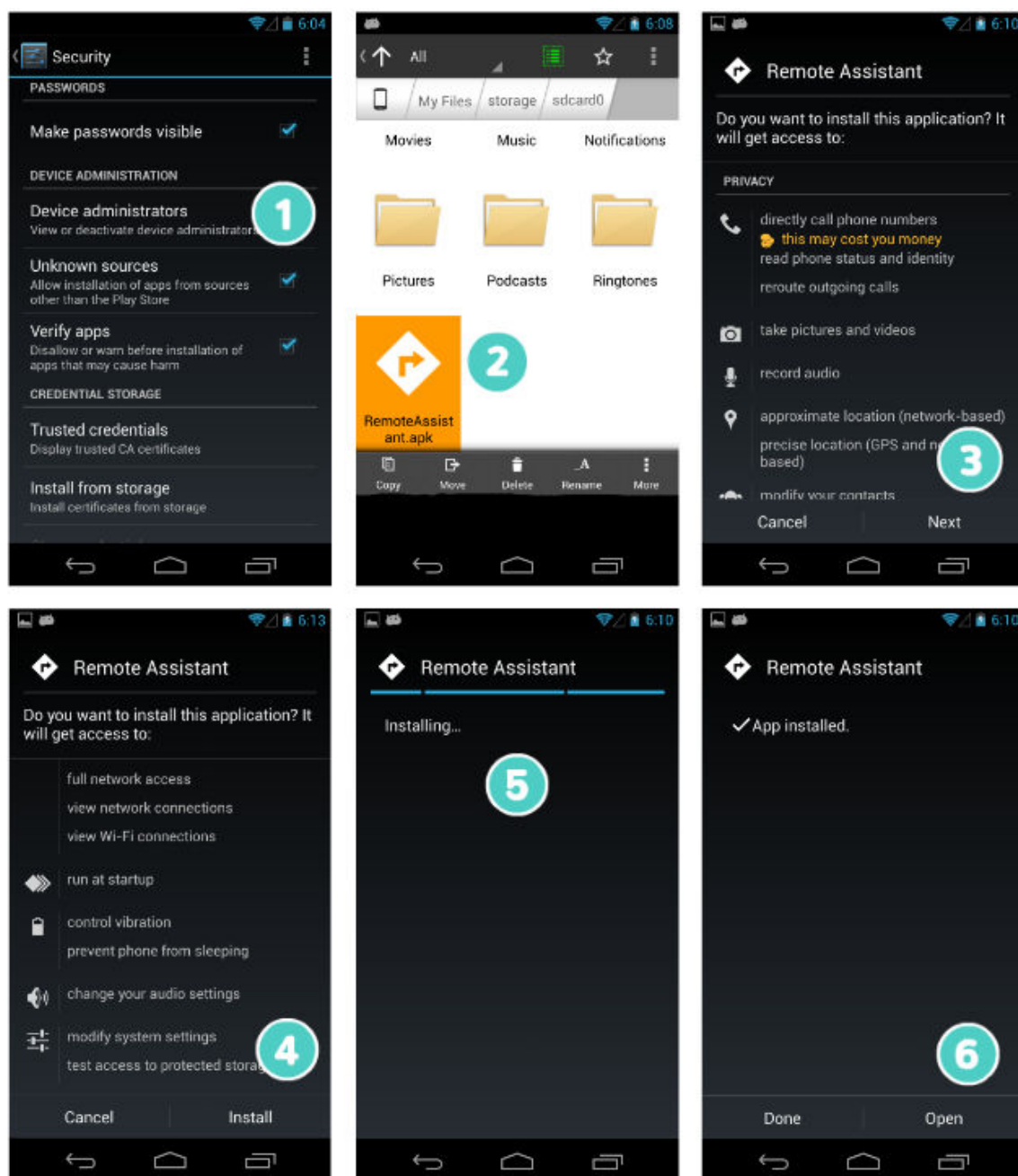
- Prítomnosť digitálneho kompasu,
- Predná alebo zadná kamera s rozlíšením aspoň 2 megapixely,
- Podpora technológií pre prenos údajov Wi-Fi, HSDPA a HSUPA, prípadne LTE,
- Aspoň 15 MB voľného miesta vo vnútornej pamäti telefónu.

8.4 Inštalácia aplikácie

Keďže aplikácia *Vzdialený asistent* v čase písania práce nebola zverejnená v obchode s Android aplikáciami Google Play Store, je pre inštaláciu potrebné skopírovať súbor *bin/RemoteAssistant.apk* z priloženého CD do vnútornej pamäte telefónu, prípadne úložiska na microSD karte. Pred započatím samotnej inštalácie je potrebné povoliť inštaláciu aplikácií z neoverených zdrojov v ponuke menu *Settings/Security* a zaškrtnúť v sekcii *Device administration* položku *Unknown sources*. Potom už stačí len pomocou obľúbeného súborového manažéra (napríklad voľne dostupný Astro z Google Play Store) kliknutím na súbor aplikácie *RemoteAssistant.apk* spustiť inštaláciu, presunúť sa na zobrazenej zhrňujúcej aktivite úplne dole a stlačením tlačidla *Install* inštaláciu dokončiť. Postupnosť týchto krokov je znázornená aj na obrázku 8–1.

8.5 Použitie aplikácie

Každý používateľ aplikácie *Vzdialený asistent* potrebuje pre vstup do aplikácie svoj účet. O tento účet je možné v tejto chvíli iba požiadať niektorého člena projektu Mapz, keďže nová platforma Mapz, o ktorej som písal v implementačnej časti diplomovej práce je zatiaľ v rannej fáze vývoja a s tým sa samozrejme spájajú aj niektoré ďalšie obmedzenia, o ktorých budem písať v kapitole 8.7.



Obr. 8 – 1 Inštalácia aplikácie *Vzdialený asistent*.

V aplikácii sa rozoznávajú tri typy používateľských účtov, pričom používateľské prostredie je pre každý typ účtu takmer rovnaké s výnimkou jedinej obrazovky, ktorá sa zobrazuje po nadviazaní VoIP spojenia medzi asistentom a nevidiacim. Táto obrazovka sa zobrazuje v dvoch rozdielnych verziách podľa typu účtu. Používateľské

účty sa delia na:

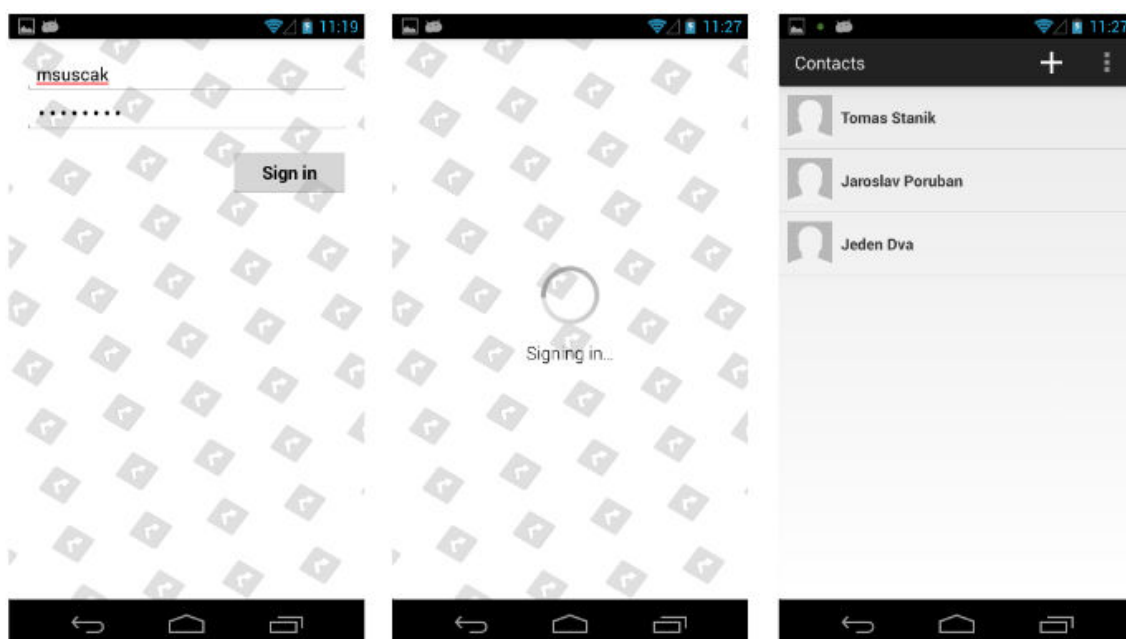
- účty pre zrakovo postihnuté osoby,
- účty pre asistentov,
- účty pre administrátorov.

8.5.1 Prihlásenie sa do aplikácie

Po získaní prihlasovacích údajov k používateľskému účtu systému Mapz je možné pristúpiť k používaniu nainštalovanej aplikácie *Vzdialený asistent* jej spustením zo zoznamu aplikácií operačného systému Android. Zrakovo postihnuté osoby musia pred prvotným spustením aplikácie *Vzdialený asistent* požiadať o aktiváciu služby *TalkBack* vidiaceho rodinného príslušníka alebo na tento účel prideleného pracovníka z projektu Mapz.

Po prvom spustení aplikácie sa používateľovi zobrazí prihlasovacia obrazovka (obrázok 8–2 naľavo), kde zadá jemu pridelené prihlasovacie údaje, ktoré predtým obdržal na požiadanie. Následne dôjde stlačením tlačidla *Sign in* k pokusu o prihlásenie, pričom sa v tej chvíli zobrazí obrazovka (obrázok 8–2 v strede), ktorá značí, že sa aplikácia pokúša o nadviazanie spojenia s autentifikačným serverom. Ak boli zadané prihlasovacie údaje správne, používateľ sa dostane do zoznamu svojich kontaktov (obrázok 8–2 napravo). Tieto prihlasovacie údaje už pri ďalšom prihlásení do aplikácie *Vzdialený asistent* nie je potrebné zadávať, keďže aplikácia si ich zapamätá a automaticky používateľa prihlási.

V čase písania práce ešte nebolo možné pridávať kontakty priamo z aplikácie *Vzdialený asistent*, pretože sa počas vývoja kladol dôraz hlavne na vytváranie VoIP spojenia pre účely vzdialenej asistencie.

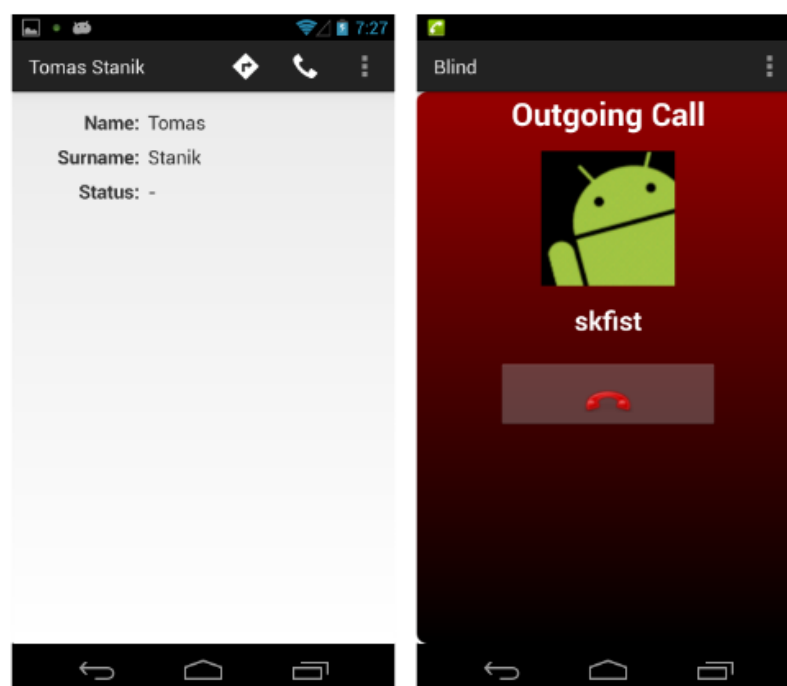


Obr. 8–2 Spustenie aplikácie a prihlásenie sa

8.5.2 Nadviazanie VoIP spojenia pre účely vzdialenej asistencie

V tejto chvíli sa už nevidiaci môže pokúsiť o nadviazanie VoIP spojenia pre účely vzdialenej asistencie v neznámom prostredí. Prejdením na niektorý kontakt zo zoznamu kontaktov pomocou špeciálnych gest sa dostane k detailom danej kontaktnej osoby (asistenta). Ukážka takejto obrazovky je na obrázku 8–3 vľavo. Následne sa nevidiaci dostane na tlačidlo, ktoré má názov *Request assistance* (na obrázku 8–3 vľavo zobrazené ako symbol šípky v hornej lište) a jeho stlačením požiada o vzdialenú asistenciu. To sa prejaví zobrazením príslušnej obrazovky, ktorá je na obrázku 8–3 vpravo.

Po prijatí požiadavky o vzdialenú asistenciu asistentom sa nadviaže VoIP spojenie a obom účastníkom sedenia sa zobrazí príslušná obrazovka. Na obrazovke, ktorá sa zobrazí nevidiacemu je len jediné tlačidlo pre ukončenie sedenia (obrázok 8–4 vľavo). Na obrazovke asistenta sa v dolnej polovici zobrazuje obraz z kamery mobilného telefónu nevidiaceho a v hornej polovici sa zobrazuje mapa, na ktorej je

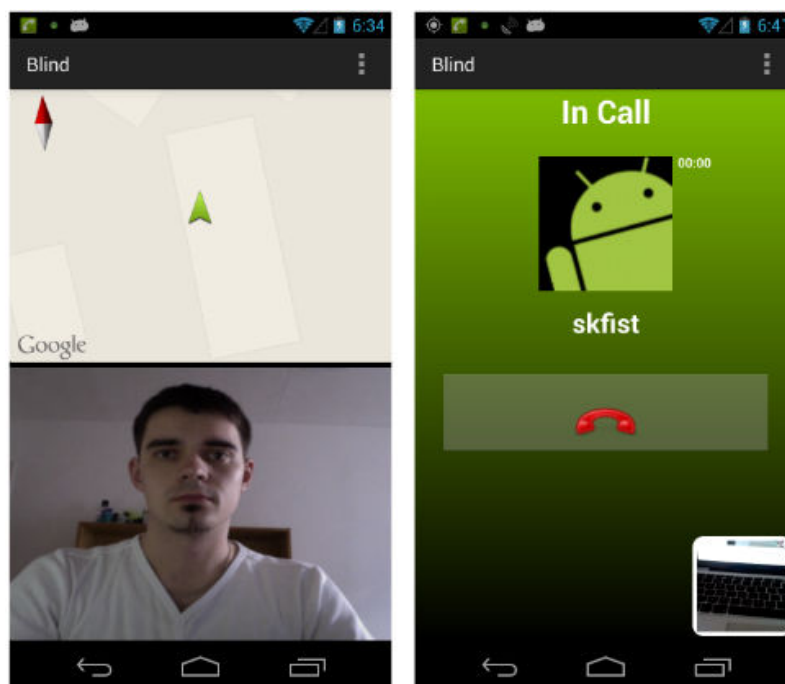


Obr. 8–3 Pokus o nadviazanie VoIP spojenia medzi nevidiacim a asistentom.

zelenou šípkou zobrazená poloha a smer natočenia nevidiaceho získané z príslušných modulov mobilného telefónu nevidiaceho (obrázok 8–4 vpravo). Asistent má teraz lepší prehľad o situácii na mieste, kde sa nachádza nevidiaci. Akonáhle sa rozhodne niektorý z používateľov ukončiť vzdialenú asistenciu, stlačí tlačidlo *End call* na zobrazenej obrazovke. Toto tlačidlo sa v používateľskom prostredí asistenta zobrazuje po kliknutí na tlačidlo so symbolom v tvare troch bodiek v hornej časti obrazovky.

8.5.3 Prepnutie účtov a odhlásenie sa z aplikácie

Aplikácia po stlačení tlačidla, ktoré slúži na prepnutie do základnej obrazovky operačného systému ostáva pre potreby akceptovania prichádzajúcich žiadostí o vzdialenú asistenciu bežať na pozadí, čo je označené symbolom v tvare zelenej guľičky zobrazenej v hornej lište. Ak sa chce používateľ odhlásiť, stlačí tlačidlo *Sign out*, ktoré sa zobrazí po kliknutí na tlačidlo so symbolom v tvare troch bodiek. Potom je používateľ vyzvaný, aby potvrdil to, že skutočne chce ukončiť aplikáciu, čo je znázornené na

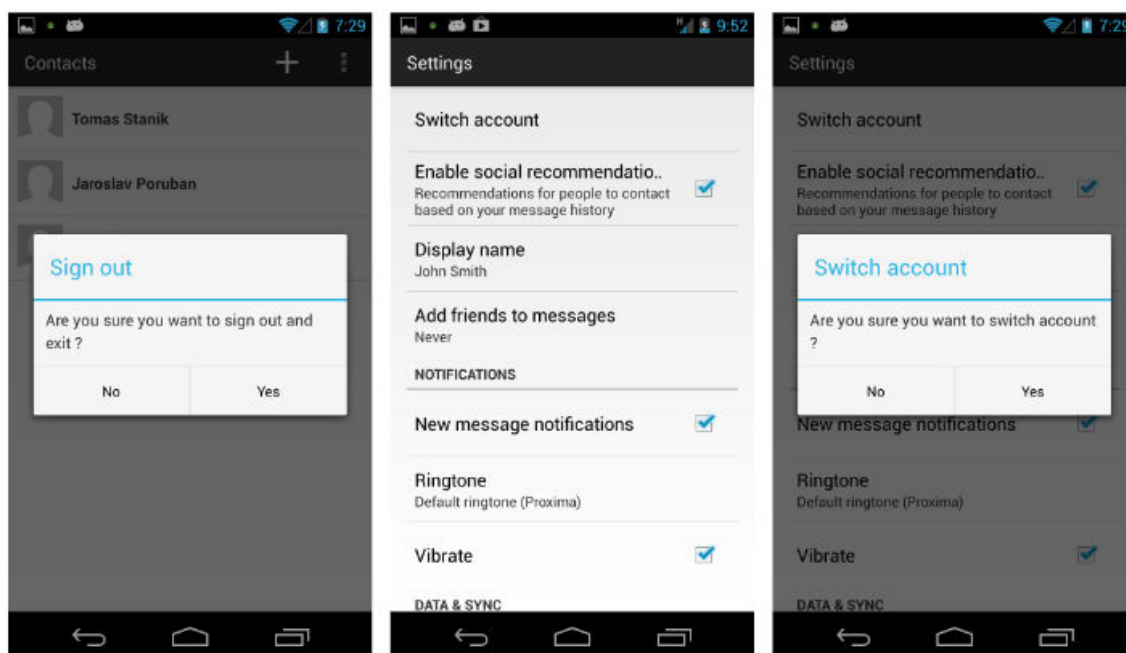


Obr. 8 – 4 Prebiehajúce VoIP sedenie medzi nevidiacim a asistentom.

obrázku 8–5 vľavo. Ak používateľ potrebuje len prepnúť účet, urobí tak prejdением do položky menu *Settings*, ku ktorej sa opäť dostane po kliknutí na tlačidlo so symbolom v tvare troch bodiek v hornej časti obrazovky a tam klikne na tlačidlo *Switch account* (obrázok 8–5 v strede). Opäť ho aplikácia vyzve pre potvrdenie toho, že sa skutočne chce odhlásiť (obrázok 8–5 vpravo).

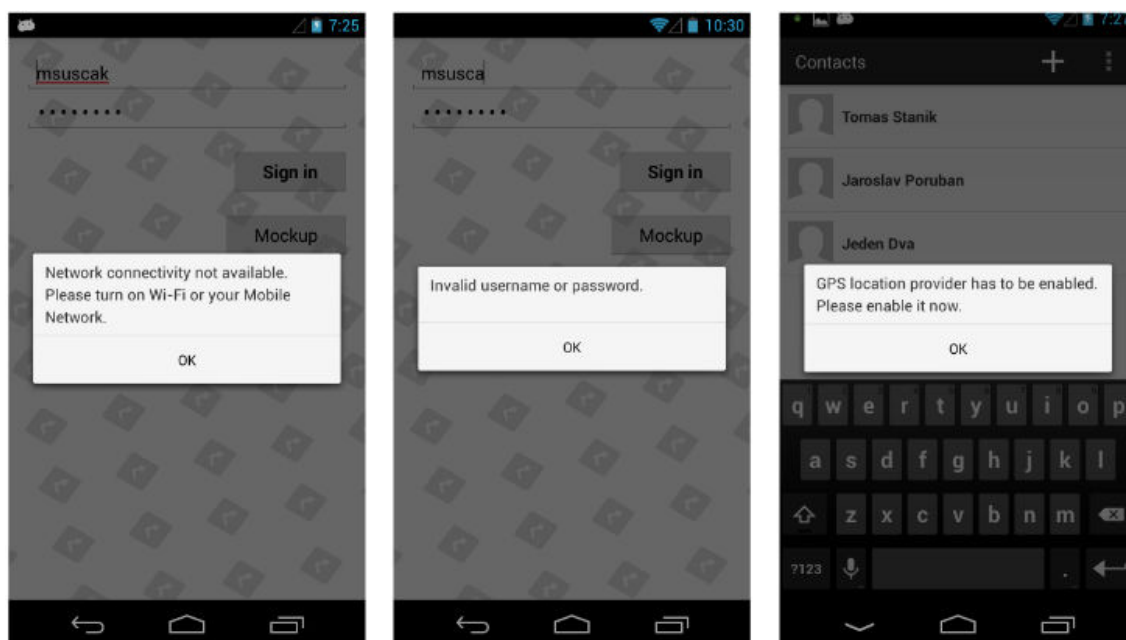
8.6 Chybové hlásenia

Aplikácia *Vzdialený asistent* oboznamuje používateľa s chybovými stavmi prostredníctvom rôznych chybových hlásení, ktoré sú zobrazené na obrázku 8–6. Chybové hlásenie z obrázku naľavo sa zobrazuje všetkým typom používateľov po spustení aplikácie, keď nie je dostupné internetové pripojenie, ktoré je samozrejme pre vzdialenú asistenciu potrebné. Ďalšie chybové hlásenie (na obrázku 8–6 v strede) sa zobrazuje používateľom keď zadajú neplatné prihlasovacie údaje od ich používateľského účtu



Obr. 8–5 Odhlásenie sa a prepnutie účtov.

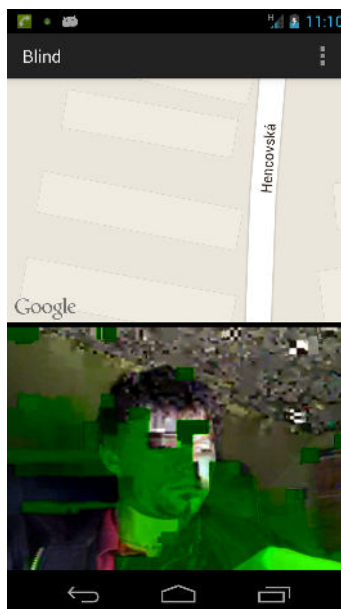
v systéme Mapz. No a posledné chybové hlásenie z obrázku napravo sa zobrazí iba zrakovo postihnutým používateľom ak nemajú aktivovaný interný GPS modul.



Obr. 8–6 Chybové hlásenia v aplikácii Vzdialený asistent.

8.7 Obmedzenia

Ako som už spomínal, systém vzdialenej asistencie Mapz spolu s aplikáciou *Vzdialený asistent* je momentálne v rannej fáze vývoja, pričom doposiaľ sa kladol dôraz predovšetkým na implementáciu pod systému určeného na nadviazanie vzdialenej asistencie prostredníctvom VoIP sedenia s rozšíreniami navrhnutými v rámci tejto diplomovej práce. Aplikácia *Vzdialený asistent* teda má niekoľko obmedzení, ktoré budú postupne odstraňované v ďalšej fáze vývoja produktu. Medzi tieto obmedzenia patrí napríklad spomínaná nemožnosť pridať kontakty priamo z mobilnej aplikácie *Vzdialený asistent*. Okrem toho nie je v tejto chvíli možné vytvoriť si účet priamo v aplikácii. Ďalším čiastočným obmedzením je problémové nadväzovanie VoIP spojenia v nespoľahlivých mobilných sieťach s veľkými výkyvmi intenzity signálu a s tým súvisiace výpadky častí obrazu pri použití nekvalitného internetového pripojenia, čo je znázornené na obrázku 8–7.



Obr. 8–7 Problémy s obrazom v prípade nekvalitného pripojenia.

Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky

Rozšírenia SIP protokolu pre účely
vzdialenej asistencie v mobilných sieťach

Príloha C: Systémová príručka

9 Systémová príručka

Táto kapitola opisuje vyvinutú mobilnú aplikáciu *Vzdialený asistent* a ním používanú natívnu knižnicu z pohľadu návrhu a architektúry. Spolu s implementačnou časťou tejto diplomovej práce by tak mala slúžiť ako referencia pre ďalšie rozširovanie vyvinutého systému vzdialenej asistencie. V úvode sa bližšie venuje analýze celého riešenia a postupom, ktoré bolo potrebné pri návrhu dodržať. Ďalej sú v samostatných podkapitolách opísané architektúry obidvoch vyvíjaných častí systému (mobilná aplikácia a natívna knižnica) a postupy potrebné pre preklad ich zdrojových kódov. Na konci príručky sú stručne spomenuté vývojové prostredia a knižnice tretích strán, použité na implementáciu riešenia.

9.1 Analýza riešenia

Aplikácia *Vzdialený Asistent* slúži predovšetkým na vytváranie VoIP spojení pre účely vzdialenej asistencie osôb v mobilných sieťach a je pevnou súčasťou navrhovanej platformy projektu Mapz. Spojenie sa zvyčajne vytvára medzi zrakovo postihnutou osobou a asistentom, ktorý môže byť rodinný príslušník alebo zamestnanec, na tieto účely zriadeného centra podpory. Počas prebiehajúceho hovoru sú asistentovi odosielané prídavné informácie o polohe a smere natočenia zrakovo postihnutej osoby.

Vrstva pre nadviazanie VoIP spojenia je založená na protokoloch rodiny SIP a vychádza z existujúcej implementácie tejto technológie v aplikačnom rámci doubango. Samotná aplikácia *Vzdialený asistent* je založená na referenčnej implementácii tohto aplikačného rámca, aplikácii IMSdroid. Rozhodnutie použiť existujúcu implementáciu bolo urobené na základe poznatkov o komplexnosti technológie SIP, získaných počas vypracovania diplomovej práce.

Podstatným faktom počas implementácie bola potreba zachovať kompatibilitu nového riešenia so záplatami aplikačného rámca doubango a aplikácie IMSdroid, vydané pôvodnými autormi. Preto bolo potrebné počas implementácie prihliadať na tento fakt a nemeniť prototypy existujúcich funkcií. Namiesto toho sa vo výraznej miere využívajú výhody zapuzdrenia nových funkcií a vlastností objektov do existujúcich tried a štruktúr aplikačného rámca.

9.2 Architektúra

Vyvinuté riešenie pozostáva z troch samostatných častí, ktorých zdrojové kódy sú na priloženom CD v priečinku *src/RemoteAssistant*. Ich vzájomný vzťah je uvedený na obrázku 9–1 a ide konkrétne o tieto časti implementovaného systému:

- Aplikácia *Vzdialený asistent* založená na aplikácii IMSdroid.
- Abstraktná vrstva rámca doubango vo forme knižnice *android-ngn-stack*.
- Modifikovaný aplikačný rámec doubango.

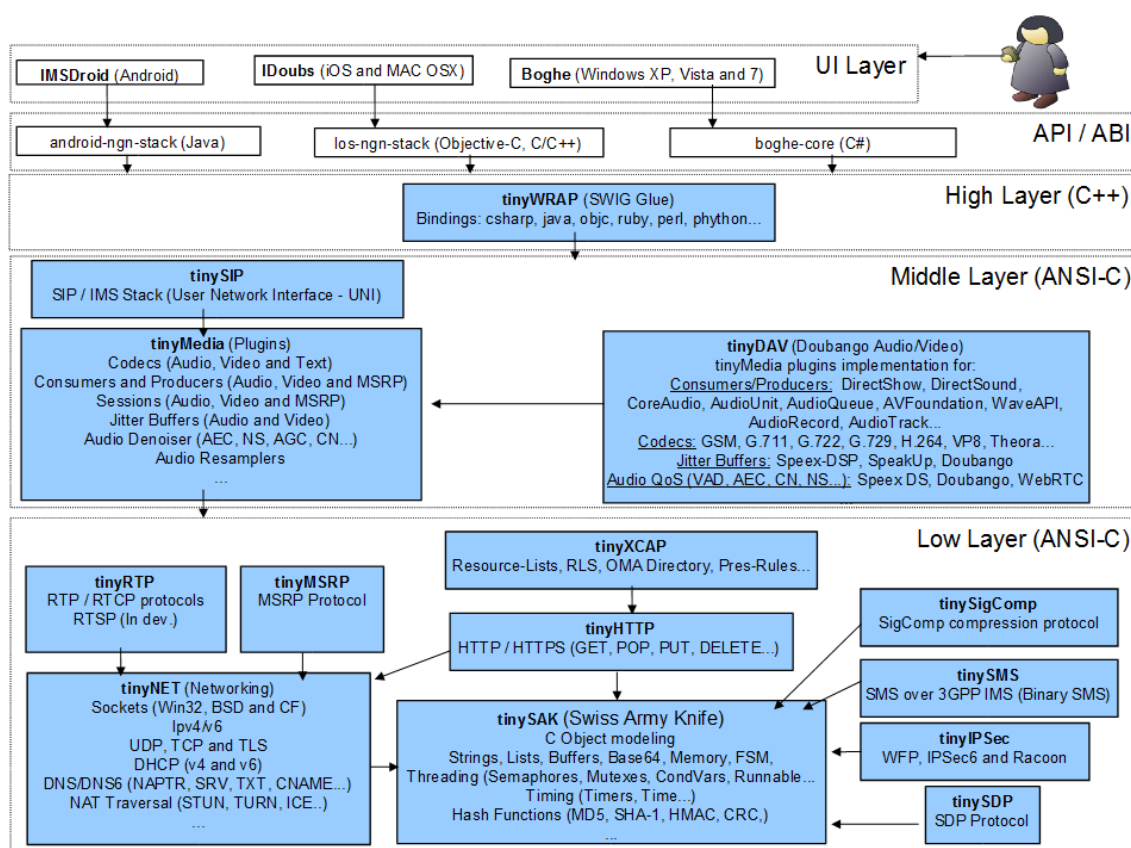


Obr. 9–1 Časti implementovaného systému.

Všetky tri súčasti budú opísané samostatne v nasledujúcich podkapitolách. Keďže ich implementácia vychádza z existujúceho riešenia aplikačného rámca doubango a aplikácie IMSdroid, dôraz bude kladený predovšetkým na vykonané úpravy.

9.3 Opis natívnej knižnice

Natívna knižnica aplikačného rámca doubango je napísaná v jazyku C kvôli použiteľnosti na rôznych typoch operačných systémov. Knižnica obsahuje implementáciu podpory pre nadväzovanie VoIP hovorov v paketovej sieti prostredníctvom technológie SIP. Skladá sa z niekoľkých samostatných balíkov, pričom každý balík zapuzdruje niektorú vrstvu implementácie. Architektúra a vzťahy medzi jednotlivými balíkmi sú znázornené na obrázku 9 – 2.



Obr. 9 – 2 Architektúra aplikačného rámca doubango.

Z obrázku by malo byť zrejmé, za ktoré funkcionality zodpovedajú jednotlivé balíky. Na priloženom CD v priečinku *src/RemoteAssistant/doubango/documentation* je možné nájsť aj oficiálnu systémovú dokumentáciu aplikačného rámca. Pre potreby budúcich úprav však uvádzam aj zoznam zmenených balíkov vrátane jed-

notlivých upravených súborov. Všetky zmenené a pridané časti zdrojového kódu sú detailne okomentované a označené komentármi pre ich ľahké vyhľadanie takto:

```
// START added by msuscak
...
/* tu sa nachadza zmeneny, resp. pridany zdrojovy kod */
...
// END added by msuscak
```

Zoznam modifikovaných súborov v jednotlivých balíkoch je uvedený pod týmto odstavcom, pričom súvisiace zdrojové súbory a hlavičkové súbory sú uvedené spolu na jednom riadku:

tinyWRAP - zo zdrojových kódov, uvedených v tomto balíku sa generuje SWIG obálka pre projekt *android-ngn-stack*, ktorá sprístupňuje a zapuzdruje všetky funkcie natívnej vrstvy.

- ActionConfig.cxx, ActionConfig.h,
- Common.h,
- MediaSessionMgr.cxx, MediaSessionMgr.h,
- ProxyConsumer.cxx, ProxyConsumer.h,
- ProxyProducer.cxx, ProxyProducer.h,
- SipMessage.cxx, SipMessage.h,
- SipStack.cxx.

tinyMEDIA - v tomto balíku boli pridané podporné štruktúry pre implementáciu signalizácie podporovaných rozšírení hlavičky RTP.

- tmedia_rtp_header_extensions.c, tmedia_rtp_header_extensions.h,

- tmedia_session.c, tmedia_session.h,
- tmedia_producer.h,
- tmedia_defaults.c, tmedia_defaults.h.

tinySIP - v tomto balíku bola pridaná podpora pre samotnú signalizáciu podporovaných rozšírení hlavičky RTP.

- tsip.c,
- tsip_action.c, tsip_action.h,
- tsip_dialog.c, tsip_dialog.h,
- tsip_dialog_invite.c,
- tsip_dialog_invite.server.c.

tinyDAV - v tomto balíku je podporný kód prenosu prídavných informácií.

- tdav_session_av.c,
- tdav_session_audio.c,
- tdav_session_video.c.

tinyRTP - do zdrojových kódov tohto balíka boli pridané štruktúry a callback funkcie, potrebné pre prenos prídavných informácií v rozšíreniach hlavičky RTP.

- trtp_manager.c, trtp_manager.h,
- trtp_rtp_header_extensions.c, trtp_rtp_header_extensions.h,
- trtp_rtcp_session.c.

tinySAK - v tomto balíku bola pridaná podpora pre zobrazovanie zaznamenaných chýb z natívnej vrstvy v aplikácii *Vzdialený asistent*.

- tsk_debug.h.

Dokumentáciu, vygenerovanú zo zdrojových kódov pridanej funkcionality je možné nájsť na CD v priečinku *doc/Generated/*.

9.3.1 Preklad natívnej knižnice

Na preklad natívnej knižnice aplikačného rámca doubango pre potreby jeho použitia v kombinácii s aplikáciou *Vzdialený asistent*, ktorá je založená na aplikácii IMSdroid je potrebné postupovať podľa oficiálnej príručky uvedenej na adrese https://code.google.com/p/imsdroid/wiki/Building_IMSDroid_v2_x a použiť namiesto oficiálnych zdrojových kódov aplikačného rámca doubango zdrojové kódy priložené na CD v priečinku *src/RemoteAssistant/doubango*. Pre tieto účely je potrebné mať stiahnuté a nainštalované Android NDK zo stránky <http://developer.android.com/tools/sdk/ndk/index.html> a základné nástroje operačného systému určené na preklad zdrojových kódov.

9.4 Opis knižnice android-ngn-stack

Knižnica *android-ngn-stack* sa na základe obrázku 9–1 z podkapitoly 9.2 používa ako integračná vrstva aplikácie *Vzdialený asistent* a aplikačného rámca doubango. Obsahuje okrem vygenerovanej SWIG obálky zo zdrojových kódov balíka *tinyWRAP* aplikačného rámca aj vyššiu abstraktnú vrstvu napísanú v jazyku Java. SWIG obálku je možné vygenerovať spustením skriptu *src/RemoteAssistant/doubango/bindings/autogen.sh* z priloženého CD, čomu musí samozrejme predchádzať stiahnutie a inštalácia aplikácie SWIG zo stránky <http://www.swig.org/>. Táto

knižnica bola súčasťou aplikácie IMSdroid a pre dosiahnutie cieľa diplomovej práce ju bolo potrebné upraviť spolu s natívnou knižnicou. K úprave došlo v týchto Java balíkoch projektu *android-ngn-stack*:

***.ngn.media** - pridaná podpora pre prenos prídavných informácií.

- NgnProxyAudioConsumer.java,
- NgnProxyAudioProducer.java,
- NgnProxyVideoConsumer.java,
- NgnProxyVideoProducer.java,
- NgnProxyVideoProducerGL.java,
- NgnProxyVideoProducerSV.java.

***.ngn.model** - pridaná trieda pre popis podporovaných prídavných informácií.

- NgnRtpHeaderExtensionDescription.java.

***.ngn.services.impl** - signalizácia podporovaných prídavných informácií.

- NgnSipService.java.

***.ngn.sip** - signalizácia podporovaných prídavných informácií.

- NgnAVSession.java.

***.ngn.utils** - callback trieda pre oznamovanie príchodu prídavných informácií.

- NgnRtpHeaderExtensionsIncomingData.java.

9.5 Opis mobilnej aplikácie

Mobilná aplikácia *Vzdialený asistent* využíva SWIG obálku a abstraktnú vrstvu v projekte *android-ngn-stack* pre prístup k natívnym funkciám aplikačného rámca doubango a pri jej vývoji bola použitá softvérová architektúra Model-View-Controller, ktorá slúži na oddelenie databázovej vrstvy, prezentačnej vrstvy a vrstvy, ktorá obsahuje riadiacu logiku aplikácie. Pôvodný zdrojový kód aplikácie IMSdroid, z ktorej aplikácia *Vzdialený asistent* vychádza bol do veľkej miery zredukovaný na minimum a triedy, ktoré boli ponechané boli výrazne modifikované. Uvádzam teda zoznam súčasných Java balíkov aplikácie *Vzdialený asistent* s ich stručným opisom, pričom ich dokumentácia sa taktiež nachádza na CD v priečinku *doc/Generated*:

- *.remoteassistant** - obsahuje všetky aktivity a fragmenty, ktoré tvoria používateľské prostredie aplikácie.
- *.remoteassistant.dialog** - obsahuje špecifické implementácie dialógov pre potvrdzovanie udalostí a oznámenia o chybách.
- *.remoteassistant.model** - obsahuje modelové triedy, ktoré sa používajú na komunikáciu s webovými službami z balíka ***.remoteassistant.rest**.
- *.remoteassistant.rest** - obsahuje jedinú triedu, ktorá predstavuje klienta, určeného na komunikáciu s webovými službami.
- *.remoteassistant.shared** - obsahuje rôzne zdieľané moduly a zoznam konštánt.
- *.remoteassistant.util** - obsahuje vlastnú implementáciu mechanizmu zaznamenávania chýb.
- *.remoteassistant.voip** - obsahuje triedy pre prácu s modulmi natívnej knižnice a Android službu, ktorá slúži na sledovanie udalostí z natívnej vrstvy aplikačného rámca doubango.

9.5.1 Preklad aplikácie

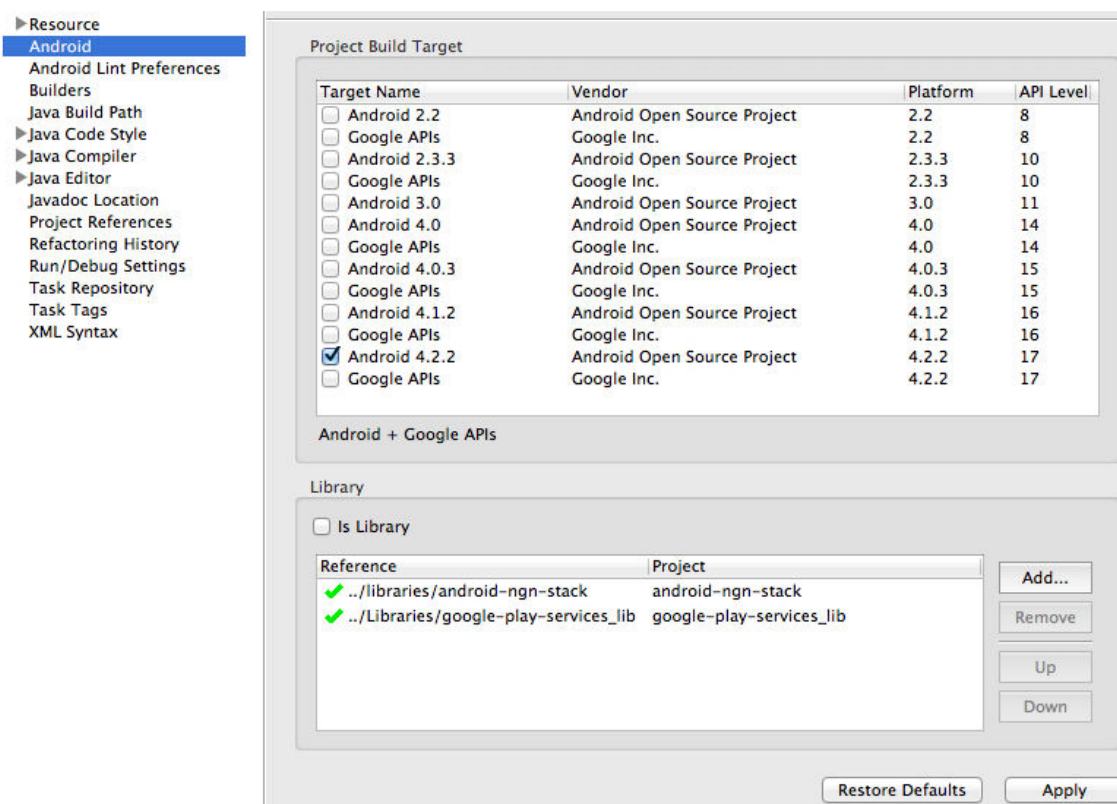
Na preklad aplikácie *Vzdialený asistent* je potrebné mať preloženú natívnu knižnicu, čo bolo opísané v stati 9.5, nainštalované prostredie Eclipse s rozšírením Android Development Tools a nainštalovaným Android SDK. Ďalej je potrebné v aplikácii Android SDK manager, ktorú je možné spustiť z prostredia Eclipse kliknutím na položku v hornej ponuke *Window->Android SDK manager*, stiahnuť SDK platformy Android 4.2.2 spolu s položkami *Google APIs* a *ARM EABI v7a System Image* ako je znázornené na obrázku 9–3.



Obr. 9–3 Stiahnutie SDK platformy Android 4.2.2.

Po vykonaní uvedených krokov je v prostredí Eclipse potrebné otvoriť tri projekty z priloženého CD, ktoré sa nachádzajú v priečinku *src/RemoteAssistant*. Jedná sa o projekty *RemoteAssistant*, *android-ngn-stack* a *google-play-services-lib*, ktoré je potrebné otvoriť súčasne, aby sa zachovali všetky väzby medzi týmito projektmi. Zároveň je dôležité sa ubezpečiť, že sa počas importovania projektov žiadna väzba nenarušila. To je možné skontrolovať vyvolaním kontextovej ponuky menu nad projektom *RemoteAssistant* a zvolením položky *Properties*, čo spôsobí otvorenie nového okna. V tomto okne je potrebné prepnúť sa do záložky *Android* a na základe obrázku 9–4 skontrolovať, či sa v spodnom okne zobrazuje zelený symbol v tvare písmena “V”. V opačnom prípade je potrebné odstrániť poškodené väzby (tlačidlo *Delete*) a pridať nové väzby (tlačidlo *Add*).

Teraz by už malo byť možné spustiť projekt *RemoteAssistant* jeho zvolením



Obr. 9 – 4 Kontrola väzieb medzi projektmi v prostredí Eclipse.

v ponuke projektov a kliknutím príslušného tlačidla *Run*, prípadne *Debug*.

9.6 Vývojové nástroje a knižnice tretích strán

Počas implementácie riešenia *Vzdialený asistent* bolo potrebné použiť rôzne knižnice tretích strán a vývojové nástroje. Vzhľadom k tomu, že aplikácia *Vzdialený asistent* a jej natívna vrstva aplikačného rámca *doubango* sú napísané v rozdielných jazykoch (Java a C/C++), boli použité dve rôzne interaktívne vývojové prostredia (IDE):

- Eclipse 4.2.2 s rozšírením Android Development Tools (ADT),
- Xcode 4.6.2.

Na dosiahnutie určeného cieľa bolo potrebné použiť tieto platformy, aplikácie a knižnice tretích strán:

- Android 4.2.2 SDK,
- Android NDK (revízia 8e),
- doubango 2.0 (revízia 850),
- IMSdroid 2.0 (revízia 550),
- android-ngn-stack 2.0 (revízia 550),
- Android Annotations 2.7.1,
- Jackson 2.1.4,
- Spring for Android 1.0.1.