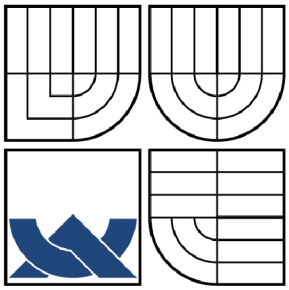
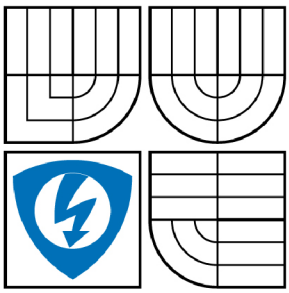


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

SIP KLIENT PRO WINDOWS MOBILE

SIP CLIENT FOR WINDOWS MOBILE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

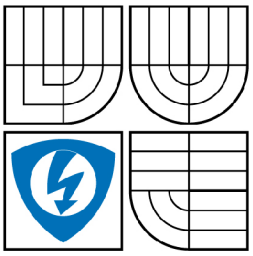
AUTOR PRÁCE — PHILIP REGUEYRA

AUTHOR

VEDOUCÍ PRÁCE — Ing. PETR ČÍKA

SUPERVISOR

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student:	Philip Regueyra	***ID:***	72913
Ročník:	3	***Akademický rok:***	2008/2009

NÁZEV TÉMATU:

SIP klient pro Windows Mobile

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se se signalizačním protokolem SIP a jeho možnými implementacemi pro využití ve VoIP. Navrhněte způsob řešení softwarového klienta v jazyce C# nebo C++ s funkčností pod operačními systémy Windows Mobile. Daný návrh realizujte. Cíl práce bude funkční SIP klient s možností obsloužit hovorové spojení typu bod-bod. K přenosu hlasu využijte dostupné kodeky a protokol RTP.

DOPORUČENÁ LITERATURA:

[1] SINNREICH, H., JOHNSTON, A. B. Internet Communications Using SIP: Delivering VoIP and Multimedia Services with Session Initiation Protocol, Canada: Wiley, 2006. ISBN 978-0471776574
[2] JOHNSTON, A. B. SIP: Understanding the Session Initiation Protocol, Second Edition. London: Artech House Publishers, 2004. ISBN 978-1580536554

Termín zadání:	9.2.2009	***Termín odevzdání:***	2.6.2009

Vedoucí práce: Ing. Petr Číka

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

ABSTRAKT

Hlavním cílem této bakalářské práce je návrh a realizace softwarového klienta, který bude schopen provádět telefonní spojení prostřednictvím paketové sítě. K tomuto účelu je využit signalizační protokol SIP společně s protokolem pro přenos multimedií RTP. Z tohoto důvodu je v bakalářské práci rámcově rozebrána struktura a fungování SIP protokolu a RTP protokolu. Jedná se o seznámení s danou problematikou, nikoli o podrobnou studii těchto nástrojů. Dále je uvedena stručná charakteristika operačního systému Windows Mobile, pro který je aplikace určena primárně. V práci je stručně popsána charakteristika několika knihoven, které byly pro použití v navrhované aplikaci uvažovány. Následuje návrh řešení, na který pozvolna navazuje rozbor realizovaného programu. V poslední části práce je popsáno fungování programu a je otestováno jeho chování v různých modelových situacích.

KLÍČOVÁ SLOVA

SIP, SDP, RTP, VoIP, klient, Windows Mobile, User agent

ABSTRACT

The main objective of this thesis is the design and implementation of client software, which will be able to conduct a phone connection through the packet network. Signalling protocol SIP together with RTP, the protocol for the transmission of multimedia, is used for this intention. Structure and functioning of the SIP protocol and RTP protocol is analyzed in this thesis. This is introducing with the issue, not a detailed study of these instruments. In addition, a brief overview of the Windows Mobile operating system for which the application is intended primarily is released. The characteristic of several libraries that have been considered for use in the application is briefly described in this thesis. Next the design of solution and the analysis of the program follow. In the last part the functioning of the program is described and its behavior is tested in various model situations.

KEYWORDS

SIP, SDP, RTP, VoIP, client, Windows Mobile, User agent

REGUEYRA, P. *SIP klient pro Windows Mobile*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 44 s. Vedoucí bakalářské práce Ing. Petr Číka.

Prohlášení o původnosti práce

Prohlašuji, že svou bakalářskou práci na téma „SIP klient pro Windows Mobile“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne ....

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Petru Číkovi za velmi užitečnou metodickou pomoc a cenné rady při zpracování práce.

V Brně dne .....

.....

podpis autora

OBSAH

Úvod **11**

1 Telefonování po Internetu – VoIP **12**

2 SIP – signalizační protokol **13**

2.1 Možnosti komunikace ... 13

2.2 Žádosti – Request messages ... 16

2.3 Odpovědi – Answer messages ... 19

2.3.1 Informační 1xx ... 21

2.3.2 Úspěch 2xx ... 22

2.3.3 Přesměrování 3xx ... 22

2.3.4 Chyba na straně klienta 4xx ... 22

2.3.5 Chyba na straně serveru 5xx ... 23

2.3.6 Globální chyba 6xx ... 23

3 Prostředí .Net a Windows Mobile **24**

4 Návrh SIP klienta pro Windows Mobile **26**

4.1 Použitá SIP knihovna ... 27

4.1.1 SIP .NET ... 27

4.1.2 Sofia-sip ... 27

4.1.3 GNU osip a eXosip ... 28

4.1.4 Open Source SIP Stack ... 28

4.1.5 nSIP a nSDP ... 28

4.1.6 PTlib a OPAL library ... 29

4.2 Návrh řešení ... 29

4.3 Struktura realizované aplikace ... 31

4.3.1 Uživatelské rozhranní MFC ... 31

4.3.2 Knihovna OPAL ... 32

4.3.3 Knihovna PTLib ... 32

4.3.4 Třídy MODIg a MOPAL ... 33

4.4 Instalace programu do zařízení ... 33

5 Test programu **34**

5.1 Spojení mezi UA v systému WM6.0 a UA v systému WinXP ... 34

5.2 Komunikace mezi dvěma UA nainstalovanými na WM6.0 38

6 Závěr **39**

Literatura **40**

Seznam zkratek, veličin a symbolů **42**

Seznam příloh **43**

A Obsah CD **44**

SEZNAM OBRÁZKŮ

Obr. 2.1: Příklad možné komunikace mezi dvěma UA zařízeními 14
Obr. 2.2: Komunikace mezi User Agenty zprostředkovaná Redirect Serverem 14
Obr. 2.3: Zprostředkování komunikace mezi dvěma UA Proxy Servery 15
Obr. 2.4: Registrace UA u Registrar Serveru 15
Obr. 4.1: Vývojový diagram při startu programu 29
Obr. 4.2: Vývojový diagram bloku „Diagram průběhu hovorů“ 30
Obr. 4.3: Schéma programu a využívání prostředků 31
Obr. 4.4: Ukázka grafického rozhraní programu ve Windows Mobile 6.0 32
Obr. 5.1: Výsledek testu – navázání spojení. 35
Obr. 5.2: Výsledek testu – přidržení hovoru. 37
Obr. 5.3: Výsledek testu – přerušení hovoru. 38

SEZNAM TABULEK

Tab. 2.1: Pole obsažená v SDP protokolu 18
Tab. 2.2: Základní skupiny odpovědí. 20
Tab. 5.1: Testy provedené pro UA ve WM6.0 a ve Windows XP 35

ÚVOD

Vzhledem k neustálému technickému rozvoji postupně dochází k transformaci telekomunikačních technologií, které v posledních deseti letech zaznamenaly největší změny. Tyto přeměny jsou motivovány požadavky na nově se vyvíjející služby a ideálním nástrojem, který je schopen nové služby adekvátně zajistit, se ukázal Internet. Vzhledem k celosvětovému rozšíření a neustále se zvyšující kvalitě připojení pod sebe umožnil zahrnout nejen telekomunikační, ale i televizní technologie. Proto se vyvíjí prostředky a nástroje, které modifikují přenos dat tak, aby jeho prostřednictvím bylo možné uskutečňovat televizní a telefonní přenosy.

Tato práce se zabývá internetovou telefonii a prostředky, které uskutečnění multimediálního přenosu umožňují. Přesto, že je vyvinuto několik standardů zajišťujících přenos hlasových a vizuálních dat prostřednictvím počítačové sítě, se tato bakalářská práce zaměřuje především na využití standardu signalizačního protokolu SIP. S tímto protokolem je spjato využití dalších nástrojů, jako jsou SDP protokol doplňující a identifikující signalizaci a RTP protokol pro přenos multimediálních dat. Právě problematika přenosu dat je blíže rozebrána v první části této práce.

Cílem práce je navržení programu pro operační systém Windows Mobile, který bude schopen navázat telefonní hovor prostřednictvím paketové sítě. K uskutečnění hovoru bude využívat signalizační protokol SIP. Výběrem správné knihovny a její implementací do vyvíjené aplikace se věnuje druhá část této bakalářské práce.

1 TELEFONOVÁNÍ PO INTERNETU – VOIP

V této práci bude pojednáváno především o internetových telefonních hovorech a konferencích označovaných anglicky jako Voice over Internet Protocol (VoIP). Jak už z názvu vyplývá, umožňují přenos hlasu prostřednictvím počítačové sítě využívající IP protokol. Nutným předpokladem pro srozumitelné a spolehlivé VoIP spojení je zajištění tzv. kvality služeb (označované jako QoS – Quality of Service). Velkou výhodou VoIP oproti klasické telefonii je jeho cena. Vzhledem k tomu, že si operátoři za volání v rámci sítě VoIP nic neúčtují, se hlavní výdaje spojené s telefonováním přesunují na poplatky za internetové připojení. Samozřejmě je potřeba zachovat kompatibilitu s klasickou telefonní sítí PSTN, k čemuž slouží VoIP brány. S nimi souvisí zbývající poplatky, které si VoIP operátoři účtují při volání do pevné a mobilní telefonní sítě. Většinou jsou však ceny stejné nebo dokonce levnější, než které nabízí klasičtí poskytovatelé.

Velmi zajímavé je využití VoIP v rámci mobilních zařízení. Není však jasné, zda dojde k masovému používání internetové telefonie přes mobilní zařízení. Hlavní překážkou je zatím nedostatečné pokrytí datovými sítěmi po celém území. Přesto však vývoj v bezdrátové oblasti probíhá. Vyvíjí se např. technologie UMA, která umožní použití přenosných zařízení jak v síti mobilního operátora, tak v internetové bezdrátové síti, pokud bude v dosahu. Tato práce se zabývá využitím mobilních zařízení pracujících na Windows Mobile komunikujících pomocí VoIP přes libovolnou wifi síť připojenou k Internetu.

Ať už budeme hovořit o VoIP telefonii využívající bezdrátové, nebo pevné připojení k Internetu, vždy zůstává zachovaný model přenosu telefonního hovoru. Jednou součástí je signalizační protokol, který zajistí vytvoření a řízení multimediální relací. Ten je však nutné doplnit o protokol pro samotný přenos hovoru. Starším komplexním řešení VoIP telefonie je H.323, standardizovaný telekomunikační organizací ITU-T. Další velmi progresivním protokolem je Session Initiation Protocol (SIP), kterým se zabýváme dále. Protože se jedná pouze o signalizační protokol, bude jej nutné doplnit o protokol zajišťující samotný multimediální přenos dat. V našem případě se bude jednat o pravděpodobně nejrozšířenější protokol RTP.

2 SIP – SIGNALIZAČNÍ PROTOKOL

SIP protokol je jedním z několika signalizačních protokolů. SIP se vyvinul jako snaha zjednodušit VoIP signalizaci a použil k tomu již zavedených principů. Roku 1999 byla definována první verze SIP protokolu v dokumentu RFC 2356. V roce 2002 byla vydána prozatím poslední verze SIP 2.0, která je specifikována v dokumentu RFC 3261. SIP protokol se stal velice populární, proto vznikaly i jiné skupiny, které vyvíjely další nástroje pracující na zdokonalení VoIP telefonie.

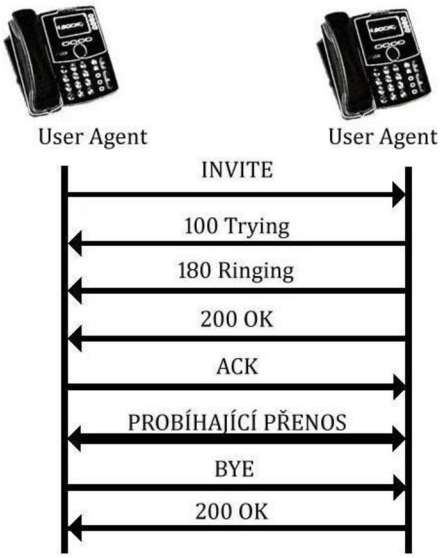
Protokol SIP pracuje na textové bázi a jeho velikou předností je jednoduchost. Vychází z praxí osvědčených protokolů HTTP a SMTP. Od HTTP si vypůjčil typ klient-server a používání URL a URI. Po SMTP zdědil uspořádání a používání hlavičky. Tyto podobnosti mu dodávají na atraktivitě. K jeho dekódování nejsou nutné žádné složité algoritmy, ale při základní znalosti kódu lze většinu podstatných informací vyčíst již na první pohled. Je založen na posílání zpráv s žádostmi (request messages) a odpověďmi (answer messages), které se od sebe odlišují zápisem. K uskutečnění spojení potřebujeme SIP klienta, který se chová jako klient a zároveň server. Samozřejmě můžeme použít i další zařízení jako servery a brány, ale v nejjednodušším případě nám pro navázání spojení postačí pouze klienti.

SIP klienti obvykle pro připojení k SIP serveru nebo k dalšímu SIP klientovi používají protokoly TCP a UDP. Podle zvoleného protokolu se poté jedná o relaci spojovanou či nespojovanou. Samotný proces přenosu zajišťuje SIP za pomoci dalších protokolů. Jedním z nich je Session Initiation Protocol (SDP), kterým jsou popsány inicializační parametry konstantně přijímaných dat při zahájení konference v ASCII kódu. Další velice podstatný protokol zajišťující přenos multimediálních dat v reálném čase je RTP a spolu s ním protokol RTCP, který zajišťuje řízení a sledování kvality přenášených dat. SIP je možné použít pro zahájení a ukončení přenosu jak v telefonních, tak ve video hovorech. Signalizační protokol má ke správné inicializaci, průběhu a ukončení relace za úkol zajistit několik základních činností. Mimo jiné musí nalézt spojení s koncovou stanicí a dále musí zjistit, v jakém stavu se koncová stanice nachází a zda používá kompatibilní prostředky přenosu (např. kodeky). Pokud dojde k navázání spojení, SIP pro přenos použije další protokoly. Během probíhající relace musí reagovat na změny vlastností a zajistit ukončení relace.

2.1 Možnosti komunikace

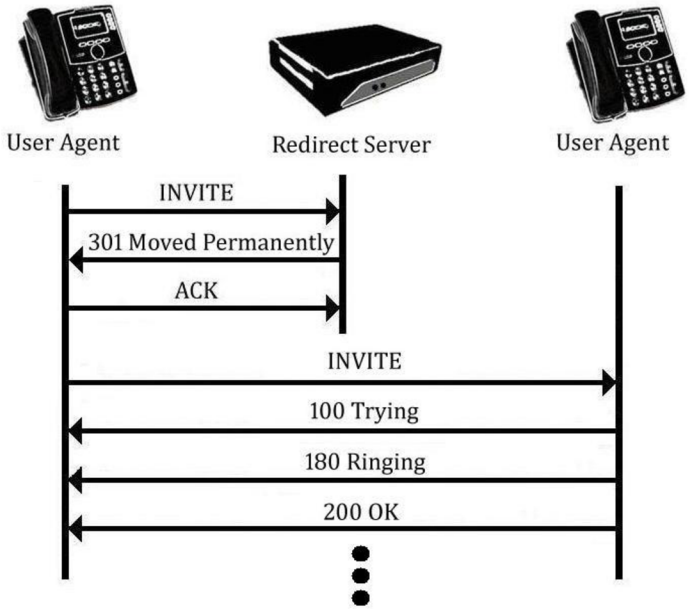
Podle doporučení RFC 3261 pro protokol SIP byly definovány základní prvky sítě. Základním prvkem je uživatelský agent, který umožňuje koncovým účastníkům sítě

obousměrnou komunikaci. Většinou se setkáme s anglickým označením User Agent (UA), jehož funkcí je vytváření a zpracování zpráv SIP protokolu. Telefonní zařízení, videotelefon, softwarový klient a jakákoli další podobná zařízení jsou pro SIP protokol chápána jako UA.

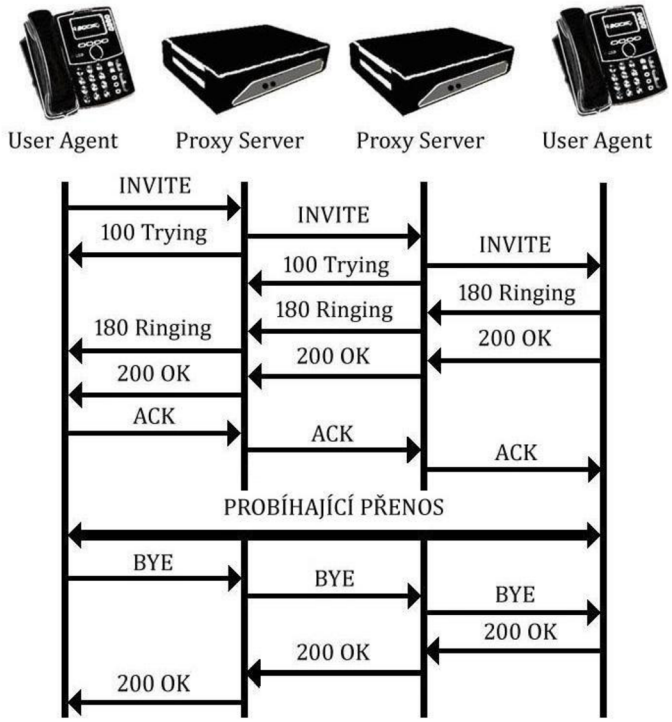


Obr. 2.1: Příklad možné komunikace mezi dvěma UA zařízeními

Podle právě probíhajícího typu komunikace se UA dělí na dvě části, a to na klientskou část (UA Client) sloužící k sestavování a řízení odchozích relací a na část serverovou (UA Server), která přijímá a řídí příchozí relace (Obr. 2.1). K tomu, aby byla zpráva od UA klienta k UA serveru úspěšně doručena, se běžně používají Proxy Servery, Redirect Servery a Registrar Servery. SIP Proxy Server (Obr. 2.3) slouží k směrování spojení, hledání účastníka v síti a zprostředkovávání propojení s dalšími sítěmi.

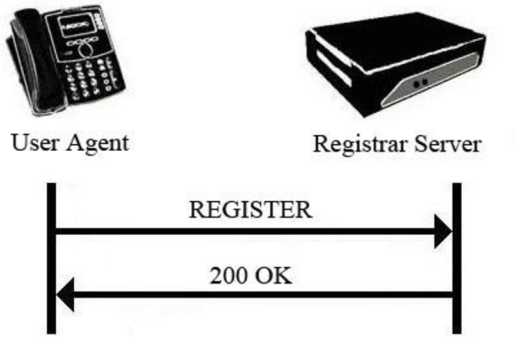


Obr. 2.2: Komunikace mezi User Agenty zprostředkovaná Redirect



Obr. 2.3: Zprostředkování komunikace mezi dvěma UA Proxy Servery

Druhým typem serveru je SIP Redirect Server (Obr. 2.2), který UA klientovi odešle zprávu s adresou požadovaného UA serveru nebo SIP serveru, který jej přiblíží směrem k cíli. Posledním definovaným prvkem je SIP Registrar (Obr. 2.4), který registruje koncové uživatele.



Obr. 2.4: Registrace UA u Registrar Serveru

V předchozím textu byly názorně zobrazeny některé možné příklady fungování komunikace s žádostmi a odpověďmi. Komplexnější přehled příkladů fungování SIP komunikace jsou blíže rozebrány v [11].

2.2 Žádosti – Request messages

Ke správě relace prostřednictvím SIP protokolu se používají zprávy žádostí. Tyto zprávy obsahují metody, z nichž základních šest je popsáno v dokumentu RFC 3261 a zbylých sedm v několika dalších RFC dokumentech. Metody jsou od zbytku zprávy odlišené velkými písmeny a nachází se na jejím začátku. Protože je SIP kód jednoduchý a snadno dekódovatelný, rozebereme si jeho strukturu a použité metody přímo na příkladu odeslaných zpráv vzorových relací. Pod prvním řádkem s metodou se zpráva dělí na jednotlivá pole začínající hlavičkou.

INVITE sip:887654@iptel.org SIP/2.0
Via: SIP/2.0/TCP 192.168.2.53:44912;branch=z9hG4bK-d87543
Max-Forwards: 70
Contact: <sip:890765@85.207.200.20:12186;transport=TCP>
To: "887654"<sip:887654@iptel.org>
From: "890765"<sip:890765@iptel.org>;tag=277ba557
Call-ID: 990aaf4a30344669MjRhYTI3MDE.
CSeq: 1 INVITE
Content-Type: application/sdp
User-Agent: X-Lite release 1002tx stamp 29712
Content-Length: 190

v=0
o=- 7 2 IN IP4 192.168.2.53
s=<CounterPath eyeBeam 1.5>
c=IN IP4 192.168.2.53
t=0 0
m=audio 13104 RTP/AVP 101
a=alt:1 2 : ThQjcxWK q67DAzpR 192.168.2.53 13104
a=alt:2 1 : B7FqqY3S zXqovyEL 85.207.200.20 12189

Zpráva začíná metodou INVITE, která slouží k sestavení relace mezi dvěma UA. Ta je následovaná volanými URI. Tímto identifikátorem je zpráva adresována UA, SIP serveru či dalšímu zařízení. V našem případě je tvar URI sip:887654@iptel.org, můžeme se však setkat i

s jinými tvary zápisu (např. sip:nick@iptel.org, sip:host@85.200.0.50, tel:+420123456789). V případě použití sip: adresy je důležité, aby byl identifikátor uváděn ve tvaru uživatel@doména. Tento tvar zápisu je obdobný jako adresování v SMTP protokolu. Jako poslední je na prvním řádku uvedeno SIP/2.0 udávající verzi SIP protokolu. Druhý řádek, označovaný jako pole Via, určuje cestu žádosti. Opět obsahuje verzi protokolu a transportní protokol (v našem případe TCP). Dále pokračuje jménem volajícího (host name) nebo (stejně jako v našem příkladě) jeho IP adresou. Tato adresa je zakončena číslem portu. Parametr branch jednoznačně identifikuje relaci. Každý SIP server, kterým zpráva projde, si přidává svoje vlastní pole Via, proto těchto polí může být ve zprávě více. Odpověď potom díky tomu k volajícímu prochází stejnou cestou zpět. Z pole Max-Forwards máme možnost vyčíst, kolika SIP servery může zpráva maximálně projít, než dorazí do cíle. V našem případě je to 70 serverů, ale jeho hodnota může být v rozsahu od 0 do 255. Každý server, který zpráva překoná, dekrementuje hodnotu o jedna. V případě, že se hodnota dostane na 0, je na tuto událost reagováno odpovídající odpovědí. Pole Contact na čtvrtém řádku určuje URI koncového uživatele, kterému náleží zpráva. Následující tři řádky s poli To, From a Call-ID jednoznačně identifikují dialog a jsou kopírovány z žádosti do odpovědi. Pole To obsahuje jméno a URI volaného účastníka umístěné do závorek typu < a >. Podobně je tomu také v poli From, které identifikuje UA, který žádost vydal (dle příkladu je jméno tohoto AU 890765 s URI 890765@iptel.org). Hodnota parametru tag je generována k dalšímu upřesnění identifikace přenosu. Funkce pole Call-ID již byla zmíněna a náš rozhovor charakterizuje řetězec 990aaf4a30344669MjRhYTI3 MDE. Po Call-ID následuje pole CSeq, které definuje pořadí transakce v dialogu. Každá další transakce zvyšuje hodnotu CSeq o jedna. Toto neplatí pouze v případě metod ACK a CANCEL. Pole Content-Type identifikuje typ média obsažených v těle SDP zprávy. Pokud zpráva obsahuje SDP část, musí být tento řádek uveden. Následuje User-Agent, kde je uvedeno, jaká aplikace je pro přenos použita. Můžeme vidět, že pro tento příklad byl použit program X-Lite. Posledním polem zprávy je Content-Lenght, které nám udává velikost SDP zprávy. Neobsahuje-li zpráva tuto část, musí být její hodnota nulová.

Pokračuje tělo SDP protokolu, které se přidává do zprávy a popisuje datový tok. Obsahuje ve svém těle informace pro sestavení spojení, informace o IP adrese a portu, na něž mají být data poslána. Dále obsahuje údaj o kodeku použitého ke kódování přenosu. Stejně jako v SIP části začínají řádky poli, tato ale začínají na malá písmena. Stručný popis použitých polí viz Tab. 2.1.

Tab. 2.1: Pole obsažená v SDP protokolu

Pole	Popis
V	Verze protokolu
O	Identifikátor relace a volajícího
S	Jméno relace
I	Informace o relaci
E	Emailová adresa
P	Telefonní číslo
C	Informace o připojení
T	Čas začátku a konce relace
R	Čas opakování
M	Informace o mediu
A	Atributy media

V případě, že se volaný UA rozhodne přijmout hovor, odešle se volanému zpráva s odpovědí 200 OK. Aby došlo k sestavení relace, je třeba odeslat žádost s metodou ACK. Když budeme uvažovat náš příklad, UA 887654 odešle UA 890765 kladné potvrzení ACK.

ACK sip:887654@iptel.org SIP/2.0
Via: SIP/2.0/TCP 192.168.2.53:44912;branch=z9hG4bK-ed43387
To: "887654"<sip:887654@iptel.org>;tag=84c32a59
From: "890765"<sip:890765@iptel.org>;tag=277ba557
Call-ID: 990aaf4a30344669MjRhYTI3MDE.
CSeq: 1 ACK
Content-Length: 0

Na začátku celé zprávy vidíme metodu ACK, která je posledním potvrzením po zprávě INVITE před samotným průběhem přenosu dat. V polích, která byla uvedena ve zprávě INVITE, má zpráva ACK stejné hodnoty. Jenom v poli CSeq je příkaz INVITE nahrazen ACK. Zpráva neobsahuje pole SDP protokolu, proto je hodnota Content-Length 0. Po bezchybném doručení této zprávy UA serveru by došlo ke spojení hovoru a výměně multimediálních dat.

Uvedli jsme si dva příklady request messages a použili v nich metody INVITE a ACK. Druhů metod je pochopitelně mnohem více. Probereme si jejich tvar a funkci. Velmi důležitou metodou je BYE. Pokud se jeden z účastníků rozhodne ukončit relaci, jeho UA odešle tuto zprávu. Po přijetí odpovědi 200 OK je přenos ukončen. Metoda REGISTER se nejčastěji používá při registraci uživatele Registrar serveru, který zaznamená uživatelské URI a poskytne ho síti, aby se vědělo, jakým směrem provádět odesílaní dat adresovaných právě tomuto UA. Zpráva CANCEL se používá ke zrušení komunikace. Na rozdíl od metody BYE, která ukončí multimediální přenos, metoda CANCEL přeruší konverzaci ještě před samotným navázáním hovoru. OPTIONS je metoda dotazující se na možnosti UA či SIP serveru. SIP Proxy nikdy negeneruje tuto zprávu. Jedná se o poslední zmíněnou metodu z šestice metod popsaných v RFC 3261. Následující metody patří do jiných RFC dokumentů. Metodu INFO používá UA k zasílání signalizačních zpráv. Odesílají se koncovému UA při probíhajícím přenosu, i když nedochází ke změně hovoru. Další možností přenosu je metoda PRACK, která slouží k dočasnému potvrzení relace. Ve stejném časovém intervalu jako zpráva CANCEL může být odeslána i zpráva UPDATE. Mění hodnoty nastávající komunikace před doručením zprávy ACK. Metoda REFFER udává novou hodnotu parametru URI, na který má být hovor směrován. Dalšími metodami jsou SUBSCRIBE a NOTIFY. Uživatel, který požaduje upozorňování na nastalou událost, použije metodu SUBSCRIBE. Na tuto žádost může reagovat jiný uživatel zprávou NOTIFY, ve které mu sdělí požadované informace. Protože SIP protokol umožňuje odesílání i Instant Messages (IM), používá k tomuto účelu metodu MESSAGE.

2.3 Odpovědi – Answer messages

Na požadavky jedné strany je potřeba reagovat adekvátní odpovědí na straně druhé. K tomuto účelu slouží odpovědi. Podrobně je problematika answer messages rozebrána v [1]. Tyto zprávy jsou odesílány k UA klientovi nejčastěji jako reakce na jeho požadavky. Jsou rozděleny do 6 základních tříd, viz Tab. 2.2.

Tab. 2.2: Základní skupiny odpovědí.

Třída	Popis
1xx	**Zprávy informační** – můžeme se setkat i s označením jako zprávy dočasné. Toto označení je dáno vlastností odpovědi, kdy UA klient už ví, že jeho zpráva byla doručena a počítá s dalšími zprávami.
2xx	**Úspěch** – tyto zprávy informují o správnosti zpracování odeslané zprávy.
3xx	**Přesměrování** – jedná se o zprávy, které určují, jakým směrem by měla zpráva dále pokračovat. Nejčastěji se jedná o odpovědi od Redirect Serverů.
4xx	**Chyba na straně klienta –** žádost není možné provést z určitého důvodu, který je dále blíže specifikován.
5xx	**Chyba na straně serveru** – při zpracování žádosti došlo k chybě podobně jako u třídy 4xx, avšak nyní na straně serveru. Většinou bývá odpověď odeslána s časovou hodnotou, kdy se může pokus o zpracování žádosti opakovat.
6xx	**Globální chyba –** zpráva je chybná a nebude ji možné přijmout nikde v síti. Odesílá ji server a podobně jako ve třídě 5xx bývá obsažen údaj o čase nevysílání.

V následujícím případě bude znázorněno, jak by vypadala odpověď 180 Ringing na výše uvedený příklad s metodou INVITE.

SIP/2.0 180 Ringing
Via: SIP/2.0/TCP 192.168.2.53:44912;branch=z9hG4bK-d87543
Contact: < sip:887654@iptel.org >
To: "887654"<sip:887654@iptel.org>; tag=84c32a59
From: "890765"<sip:890765@iptel.org>;tag=277ba557
Call-ID: 990aaf4a30344669MjRhYTI3MDE.
CSeq: 1 INVITE
Content-Length: 0

Na první pohled je zřetelná určitá podobnost s původní zprávou INVITE. Na prvním řádku však hned dochází ke změně. Začíná se verzí protokolu a až potom následuje označení odpovědi. K další změně došlo také v řádku pole Contact, kde se vyměnil volající za volaného a doplnil se údaje tag v poli From. Dále chybí část SDP protokolu a tím došlo ke ztrátě či

změně odpovídajících polí v SIP protokolu. V případě, že by zpráva došla k volajícímu UA, sdělila by mu, že koncové zařízení vyzvání a je třeba vyčkat na další odpověď.

SIP/2.0 200 OK
Via: SIP/2.0/TCP 192.168.2.53:44912;branch=z9hG4bK-d87543;received=192.168.2.53
Contact: < sip:887654@iptel.org>
To: "887654"<sip:887654@iptel.org>; tag=84c32a59
From: "890765"<sip:890765@iptel.org>;tag=277ba557
Call-ID: 990aaf4a30344669MjRhYTI3MDE.
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length:185

v=0
o=- 2 7 IN IP4 85.207.98.20
s=<CounterPath eyeBeam 1.5>
c=IN IP4 85.207.98.20
t=0 0
m=audio 13104 RTP/AVP 101

Tato zpráva potvrzuje kódem 200 OK přijetí hovoru volaným UA. Její tvar má podobný formu jako samotná zpráva INVITE a nalezneme podobnost i s odpovědí 180 Ringing. Opět došlo k připojení i SDP části.

Některé podkategorie tříd answer messages si blíže probereme dále.

2.3.1 Informační 1xx

Zpráva, která je odeslána ihned po doručení zprávy INVITE, má označení 100 Trying. Při průchodu Proxy Serverem se tato odpověď odešle zpět hned, jakmile server zprávu odešle dál. S odpovědí 180 Ringing jsme se již setkali v předešlém příkladě. Je to zpráva s oznámením vyzvánění na straně volaného klienta. V případě, že dojde k okamžitému přijetí, zpráva se přeskočí a rovnou se odešle odpověď 200 OK. Upozornění na přesměrování volání je zajištěno zprávou 181 Call Is Being Forward. Odesílá se hlavně z důvodu přípravy volajícího na delší dobu prodlevy kvůli přesměrování. Označení 182 Call Queued udává, že zpráva INVITE byla přijata a že je účastník zařazený do fronty, než bude požadavek dále zpracován.

Odpověď Session Progress s číslem 183 informuje o stavu hovoru. Na rozdíl od 100 Trying upozorní na přijetí zprávy INVITE právě tehdy, kdy k hovoru nemůže dojít.

2.3.2 Úspěch 2xx

Kód 200 OK odesílá UA, který oznamuje správné zpracování požadavku. Pokud UA obdrží v pořádku zprávu, ale není možné její provedení, odešle odpověď 202 Accepted.

2.3.3 Přesměrování 3xx

Z polí Contact příchozí zprávy si UA vytvoří algoritmus, při kterém bude kontaktovat jednotlivé URI v přesném zadaném pořadí. K této situaci dochází, když je daný požadavek v síti schopno zpracovat více URI. Touto odpovědí je 300 Multiple Choices. Odpovědí 301 Moved Permanently. Redirect Server je odesláno upozornění na permanentní změnu URI adresy požadovaného UA (Obr. 2.2). Podobnou funkci má také odpověď 302 Moved Temporarily, kdy změna dostupnosti na jiné adrese požadované URI je pouze dočasná. Odpověď 305 Use Proxy přichází společně s kontaktem na Proxy Server, který zjistí, kde se hledané URI nachází.

2.3.4 Chyba na straně klienta 4xx

Vzhledem k tomu, že se do kategorie 4xx řadí mnohem větší množství odpovědí než u ostatních tříd, nebudou uvedeny všechny. Mezi základní určitě patří 400 Bad Request, při kterém server nerozuměl požadavku. Bývá použit, když UA server několikrát obdrží zprávu INVITE se stejným Call-ID. S kódy 401 Unauthorized a 407 Proxy Authentification Required se nejčastěji setkáme, když se proxy dožaduje autentizace. V prvním případě je autorizace požadována jenom pro některé služby a v druhém pro veškerou komunikaci přes Proxy Server. Do budoucna se počítá s využitím odpovědi 402 Payment Required na zpoplatněné služby. Kód 404 Not Found se použije, když není jisté, že URI náleží některému UA. Indikace 405 Method Not Allowed je použita, když je zpráva přijata a v pořádku přečtena, ale z nějakého důvodu nebude žádost zpracována. Událost může nastat u metody REGISTER. Zpráva 406 Not Acceptable je stejně jako v předešlém případě přijata, ale požadavky nejsou pro UA korektní. Často se můžeme setkat s odpovědí 408 Request Timeout, kdy je spotřebován veškerý čas pro zpracování požadavku. Není stanoven čas, po který by žádost neměla být opakovaná. V příchozí zprávě 421 Extension Required se dozvíme, že chybí některá část hlavičky, kterou server požaduje. Ve zprávách se můžeme setkat také s kódem 483 Too Many Hops v případě, že zpráva prošla příliš mnoha servery a nebyla dodržena ta

podmínka, že pole Max-Forwards je větší než 0. Jako poslední z třídy chyb na straně klienta uveďme kód 486 Busy Here. UA není schopný na tomto umístění přijmout hovor. Jedná se o podobnou událost jako při zaznění tónu obsazeno v klasické telefonní síti.

2.3.5 Chyba na straně serveru 5xx

Základní kód této třídy je 500 Server Internal Error. Na straně serveru došlo k druhu chyby a zpráva ošetřuje dobu, po kterou by UA klient neměl opakovat žádost. Bývají obsaženy i odpovědní fráze, podle kterých je možné chybu identifikovat. Odpověď 501 Not Implemented se odešle, když server nemůže zpracovat požadavek, který jím není podporován. Kód 502 Bad Gateway odesílá Proxy Server, který je použit jako brána do dalších sítí. Z důvodu nastalé chyby v této síti je zamezeno provedení žádosti. S označením 503 Service Unavaible se můžeme sekat, když požadovaná služba na serveru není dostupná. K opakování požadavku může dojít během několika mála sekund. Zprávu 504 Gateway Timeout odesílá stejně jako v případě 502 Bad Gateway Proxy Server sloužící jako brána do další sítě. Je to reakce na nedodržení časového limitu s čekání na odpověď druhou sítí. Další odpověď prozatím není aktuální, protože se používá verze protokolu SIP 2.0. Jedná se o 505 Version Not Supported pro nekompatibilitu verzí. Poslední zprávou z této skupiny je 513 Message Too Large, kdy UA server oznámí, že zprávu nemohl zpracovat pro její přílišnou délku.

2.3.6 Globální chyba 6xx

Zástupců této skupiny není mnoho. Začněme 600 Busy Everywhere, která je definitivní verzí odpovědi 486 Busy here. V případě, že neexistuje možnost zpracování požadavku jinou sítí, se nemá request message opakovat. Kód 603 Decline má stejný důsledek jako 600 Busy Everywhere, jenom ve zprávě není uvedeno, z jakého důvodu došlo k zamítnutí. Obdobou odpovědi 404 Not Found je zpráva 604 Does Not Exist Anywhere. V tomto případě se však ví, že požadovaný uživatel není nikde v síti. Na závěr uveďme zprávu označenou 606 Not Acceptable. UA sever ji odešle v případě, že některou z vlastností relace není možné akceptovat a relaci zahájit. Požadavek může být opakován s různými nároky na multimediální přenos, kdy některý z nich může být pro UA server přijatelný.

3 PROSTŘEDÍ .NET A WINDOWS MOBILE

Společnost Microsoft vyvinula pracovní prostředí .NET jako reakci na svou dlouholetou snahu zachovat zpětnou kompatibilitu svých operačních systémů. Už od verze Windows 3.1 z roku 1992 je k vytváření aplikací používáno stejné rozhraní. Toto rozhraní se označuje API a nové požadavky na software a hardware měly za následek jeho rozšiřování. Každým vývojem se k rozhraní API přidávaly nové funkce a možnosti. Tímto modelem postupného vývoje bylo zajištěno, že většina softwaru pracujícího pod jedním operačním systémem pracovala i pod systémem dalším. Jako velmi podstatná nevýhoda tohoto způsobu upgradu se ukázala jeho vždy o něco složitější struktura. Tato situace Microsoft přiměla k vytvoření nástroje, který by dosavadní funkce zjednodušil a zároveň podporoval nový hardware. Tímto prostředím se stalo .NET s nově vytvořeným jazykem C#. V [8] podrobněji popsána historie vzniku prostředí .NET.

Platforma .NET je knihovna, která je rozsáhlá a kompletní stejně jako bylo stávající rozhraní Windows API. Kromě těchto možností v sobě ještě zahrnuje možnost využívání v oblastech, jako jsou práce s databázemi, připojení k Internetu a využití webových služeb. Prostředí .NET funguje pouze pod systémem Windows, protože zatím nebylo navrženo pro práci s jinými systémy. Prostředí .NET můžeme chápat jako prostředníka mezi operačním systémem Windows a dalšími aplikacemi.

Základní komponentou potřebnou pro běh aplikací pod tímto prostředím je Microsoft .NET Framework. Nabízí rozhraní pro spouštění takto vytvořených aplikací a poskytuje dostupné knihovny. Vývoj se však nezastavil a vyšlo několik verzí .NET Frameworku. Verze 1.0 byla vydána roku 2002 a verze 1.1 byla jejím vylepšením. K podstatné změně došlo u vydání 2.0 z roku 2005, kdy se změnilo jádro pro psaní programů. Prozatím nejaktuálnější je verze 3.5. Ke každému Frameworku bylo vydáno také vývojové prostředí Visual Studio k tvorbě programů.

Společností Microsoft nebyl striktně určen jazyk, kterým by se aplikace musely vytvářet, avšak jazyk C# je vytvořený od základu právě pro prostředí .NET. Tento jazyk byl vyvinut k maximálnímu využití všech výhod nového vývojového prostředí. K vytváření aplikací ale podmínka znalosti nového jazyka C# není zapotřebí. Společnost Microsoft rozšířila velmi populární a rozšířený jazyk C++ pro práci s Frameworkem. Podobně to pro uživatelský komfort udělal i s jinými jazyky, takže se můžeme setkat s jazyky Visual Basic .NET, J#, Delphi, Fortran a dalšími.

Windows Mobile je operační systém spojený s přenosnými zařízeními jako PDA, smartphony a dalšími [14]. Jeho hlavním účelem je ovládání uvedených zařízení a úplné využití všech jejich možností. Tím částečně zastupuje funkce klasických stolních počítačů a notebooku a zužitkovává přednosti klasických mobilních telefonů. První verzí tohoto operačního systému byl Pocket PC 2000. Zatím posledním operačním systémem je Windows Mobile 6.1. Pro tento kompaktní operační systém byly vyvinuty speciální verze Frameworku, aby i na těchto zařízeních bylo možné využít výhod nového prostředí. Frameworky byly označeny příhodným názvem .NET Compact Framework. První verze byla vydána roku 2002 a zatím poslední verze 3.5, stejně jako u klasického Frameworku, na začátku roku 2008. .NET Compact Framework používá některé knihovny tříd shodné s klasickým .NET Frameworkem a některé knihovny přímo určené pro mobilní zařízení (např. Windows CE InputPanel).

4 NÁVRH SIP KLIENTA PRO WINDOWS MOBILE

Přestože cílem této bakalářské práce je softwarové řešení klienta pro platformu Windows Mobile, samotné jeho vyvíjení podle informací od společnosti Microsoft by se nemělo příliš lišit od vývoje obdobného UA pro běžný operační systém Windows. Stejné nástroje a knihovny jsou z veliké části společné pro systém Windows a jeho kompaktní verzi Windows Mobile. Proto byl vývoj plánován nejprve pro prostředí Windows XP a následně jeho zpracováním pro prostředí Windows Mobile 6.

Velmi podstatnou otázkou při navrhování UA bylo rozhodnout, jaký programovací jazyk zvolit. Protože tato selekce dále ovlivňovala výběr použitých volně stažitelných programů, byla volba mezi jazyky C++ či C# dosti důležitá. První ze jmenovaných, jazyk C++, je vhledem ke své delší existenci nástrojem velmi zajímavým hlavně díky možnosti využití velkého množství vyvinutých knihoven pro tento jazyk. Vzhledem k tomu, že projekt je principielně určen pro zařízení Windows Mobile, se jako nejvhodnější jevilo použít jazyk C# přímo vyvinutý společností Microsoft. Jedná se o programovací jazyk sice mladší, ale od svého vzniku podporovaný společností Microsoft. Proto se programová základna rozrůstá velice rychle a je možné objevit knihovny zabývající se VoIP telefonií, které jsou šířeny s otevřeným zdrojovým kódem.

Jako vývojové prostředí bylo zvoleno Visual Studio od Společnosti Microsoft. Nejnovější verzí je Visual Studio 2008, avšak vhledem k bezproblémové zpětné kompatibilitě byla zvolena verze Visual Studio 2005. Microsoft na svých internetových stránkách věnuje obsáhlou podporu jednak k samotnému programu, a jednak ke knihovnám dodávaným společně s ním v balíčku.

Na Internetu existují různé druhy hotových řešení funkčních SIP klientů. Podstatně se liší vlastním zpracováním a funkcemi. Není však mnoho těch, které by měly přehledně řešeno rozhraní pro ovládání programu. Pro operační systém Windows Mobile bylo takových programů ještě méně.

Vzhledem ke skutečnosti, že pravděpodobně neexistuje zajímavé řešení SIP klienta pro Windows Mobile, se ukázalo jeho vytvoření od základu jako nejschůdnější. Návrh bude využívat SIP knihovnu, která bude zajišťovat spojení s ostatními klienty. Při realizaci návrhu bude snaha o co nejjednodušší, přitom ale přehledné ovládání. To bylo největším nedostatkem již zmíněných projektů.

4.1 Použitá SIP knihovna

Selekce správné SIP knihovny byla jedna z nejdůležitějších kroků návrhu SIP klienta. Pochopitelně by byla možnost vytvořit si knihovnu vlastní. Mělo by to své výhody, jako dokonalá znalost navržených metod a její snazší implementace do programu. Na stranu nevýhod se však staví velmi podstatná časová náročnost navržení takovéto SIP knihovny. Deklaraci knihovny by nebylo možné splnit do termínu stanoveného pro tuto práci. Proto se ukázalo jako efektivnější vyhledat si knihovnu na Internetu a seznámit se s jejími možnostmi použití. Většina serverů označuje tyto knihovny anglickým Stack, proto je toto označení použito i v této práci. Do češtiny by se tento výraz dal přeložit jako skladiště. Z různých SIP Stacků nacházejících se volně ke stažení na Internetu bylo potřeba vybrat ten, který bude nejvhodnější. Převážná většina měla obsažen naimplementovaný SDP protokol, k některým však bylo nutné použít samostatný SDP Stack. RTP protokol ve své struktuře neobsahoval žádný ze SIP Stacků. V tomto případě bude tedy potřeba použít samostatný RTP Stack. Následuje seznam SIP Stacků, ze kterých bylo vybíráno.

4.1.1 SIP .NET

Jedná se o SIP Stack od společnosti Independentsoft vyvinutý pro .NET Framework a .NET Compact Framework. Tím je zajištěná slučitelnost s operačním systémem Windows Mobile. Celá knihovna je psána v programovacím jazyce C#, což zaručuje plnou kompatibilitu všech tříd a metod s již zmiňovanou platformou Microsoft .NET Framework a Microsoft .NET Compact Framework. Její veliká přednost je možnost použití pro všechny zatím vyvinuté verze této platformy. Podle údajů udávaných společností by měla zařídit veškeré funkce SIP protokolu a SDP protokolu. Jediné, co knihovně chybí, je RTP Stack, který by se však dal vyřešit doplněním některým externím. Podle uvedených údajů by se zdálo, že se jedná o ideální SIP Stack, který se pro tuto práci dá použít. Jeho největší nevýhodou je druh licence, která umožňuje jeho použití jen po dobu jednoho roku, a to za nepřiměřeně vysokou cenu. Pro aplikaci klienta knihovna nebude použita.

4.1.2 Sofia-sip

Sofia-sip je open-source SIP UA knihovna, která je navržena podle specifikace IETF RFC3261. Je napsána v C, vyvíjena hlavně pro Linux a licencována pod GNU Lesser General Public License (L-GPL). Kromě základních možností SIP protokolu podporuje také SIMPLE pro použití instant messagingu (IM). Stejně jako předchozí protokol obsahuje SDP knihovnu, ale RTP Stack jí chybí. Přesto, že je tuto knihovnu možné použít volně, je pro práci

nevyhovující z důvodu, že je napsaná v jazyce C. Programovací jazyk C# je od jazyka C tak odlišný, že pokusy o převod do kompatibilního jazyka by byly velice obtížné.

4.1.3 GNU osip a eXosip

Obě knihovny vycházejí z komerční verze SIP Stacku Antisip, avšak licence GNU osip je L-GPL a eXosip GPL. Oba Stacky splňují všechny předpoklady k plnohodnotnému navázání multimediálního přenosu, i když podle dostupných informací byly určené pro použití do programů založených na komunikaci bod-bod. GNU osip má malou velikost a pracuje na nižších vrstvách. Knihovna eXosip je novou knihovnou vycházející z GNU osip již pracující i na vysokých vrstvách, což umožňuje snazší použití pro realizaci SIP UA. Obě knihovny neobsahují RTP Stack a stejně jako v předešlém případě je zde velkou překážkou jejich vytvoření v programovacím jazyku C. Velmi zajímavá je obsáhlá dokumentace k SIP Stackům. Z důvodu nevhodného programovacího jazyka knihovna nebude použita.

4.1.4 Open Source SIP Stack

Nejedná se zcela o klasický SIP Stack, ale o komunitu lidí, kteří vyvinuli UA aplikaci OpenSBC a OSSPhone. Jsou to jednoduché aplikace, které se podle podkladů na jejich internetových stránkách dají snadno rozšiřovat a měnit. Různé verze těchto programů obsahují odlišné funkční SIP knihovny a jsou navrženy jak pro práci pod Windows, tak i pod Linux. Kvůli většímu množství používaných SIP Stackům se můžeme setkat se třemi druhy licencí – Mozilla Public License (MPL), GPL a L-GPL. Verze jsou vytvořeny jazykem C++, což by umožňovalo použití s jazykem C#. Ze SIP Stacků nebylo možno vybrat, protože vyhledávaní vhodné knihovny SIP by mezi různými verzemi programů OpenSBC a OSSPhone zabralo příliš mnoho času. Navíc by nejspíš nebylo jednoduché vyjmout části SIP Stacku tak, aby byla zachována funkčnost.

4.1.5 nSIP a nSDP

Tento open-source projekt vznikl k použití v .NET aplikacích verze 2.0. Obsahuje zvlášť SIP Stack a SDP knihovnu, proto je potřeba k nim přistupovat jednotlivě. Jeho velkou předností je vývoj v jazyce C#. Autoři na svých stránkách slibují bohatou dokumentaci [4]. V projektu knihovny byla přímo vytvořena zvláštní třída, kde je možné testovat funkčnost všech metod. Další výhodou je variabilita při používání obou součástí, která by mohla usnadnit návrh celé aplikace. Proto tato knihovna byla vybrána při prvním vývoji programu. Pro využití pod Windows Mobile by to znamenalo nutnost výrazného zásahu do celé její

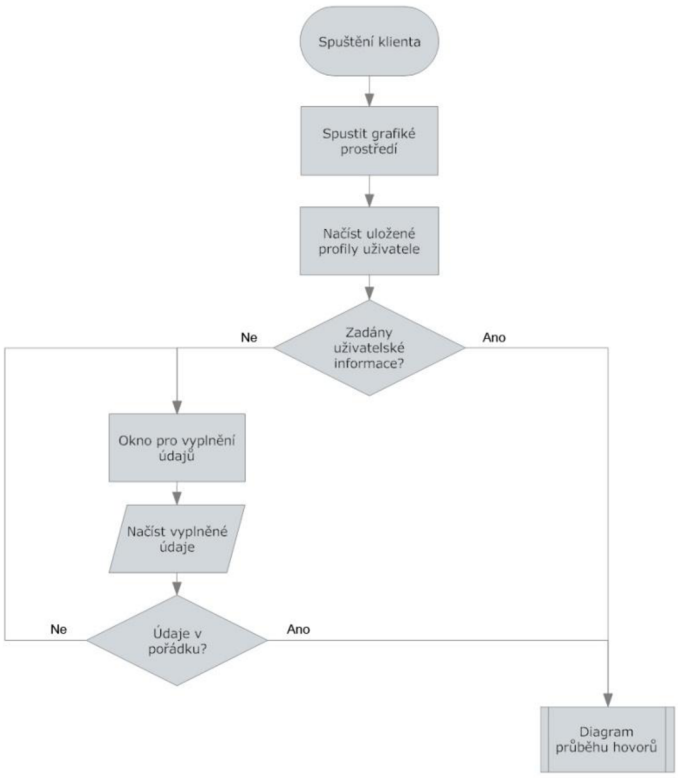
struktury, což by bylo časově velice náročné a přesahovalo to dobu určenou pro tuto bakalářskou práci.

4.1.6 PTlib a OPAL library

Posledními, a tím i dodatečně uvažovanými knihovnami, byly PTlib a OPAL library. Knihovny vyvinula společností RSDevs, kterou jsou také využívány [12]. Jedná se o knihovny obsahující rozsáhlou SIP knihovnu, která je navržena pro spolupráci s operačním systémem Windows Mobile verze 5.0 i 6.0. Programovacím jazykem je C++, kterým se k jednotlivým metodám přistupuje. Předností těchto knihoven je již zabudovaný RTP Stack a dále podporují několik typů kodeků jako G.711, Speex, H.261. Jelikož jsou poskytovány s licencí MPL, knihovny byly vybrány k použití při realizaci UA klienta.

4.2 Návrh řešení

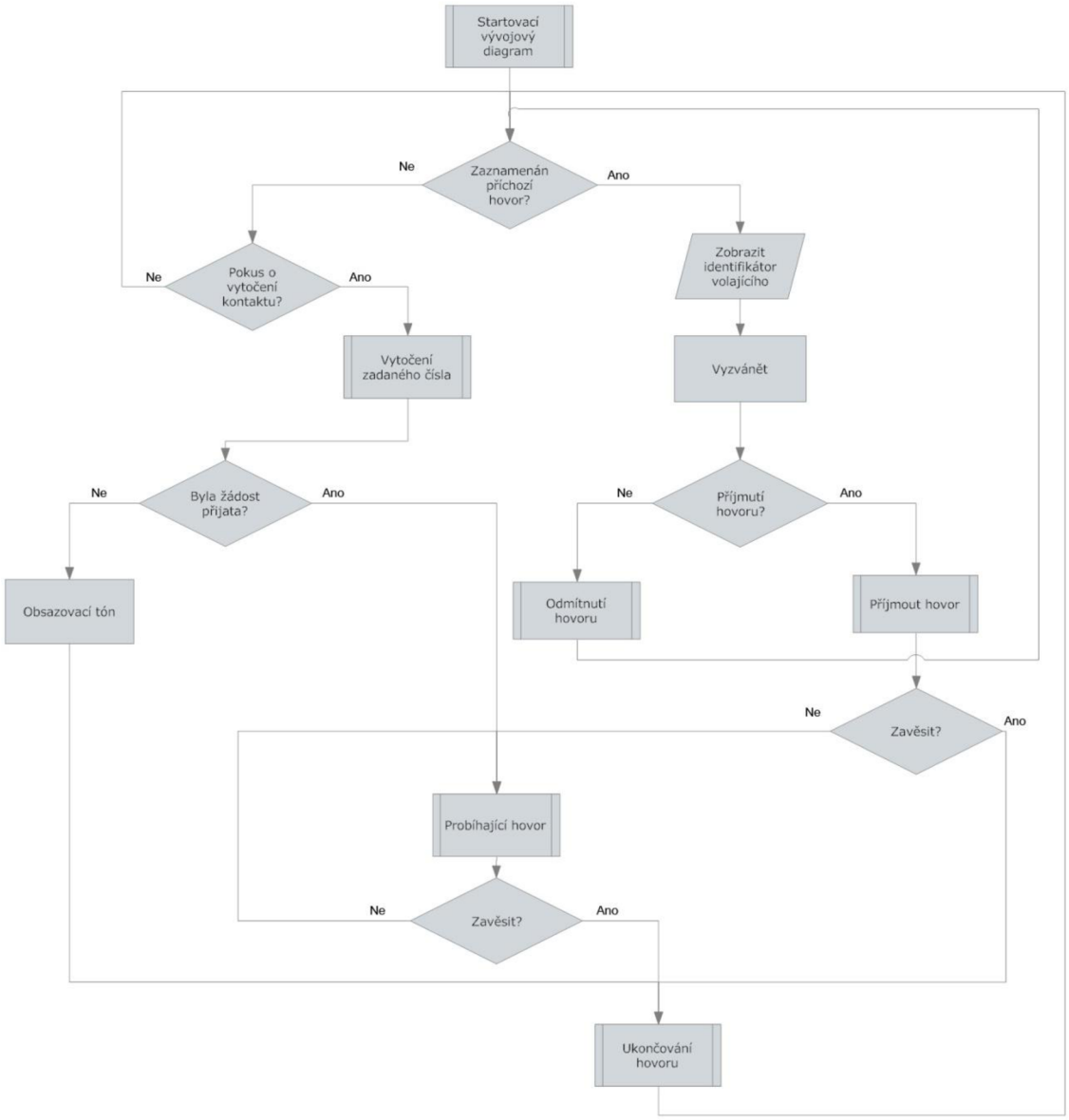
Při návrhu projektu byl vytvořen obecný vývojový diagram (Obr. 4.1), podle kterého by obecně měl program pracovat.



Obr. 4.1: Vývojový diagram při startu programu

Jedná se o vývojový diagram při spuštění programu. Při spuštění UA je zavedeno grafické prostředí, které slouží k interakci mezi uživatelem a kódem. Po provedení tohoto bodu jsou načteny údaje, které se zachovaly v paměti z předchozí práce s programem (např. volaná IP

adresa). Další část diagramu je plánována pro další možnost rozšíření UA, kdy jsou pro registraci pod některý ze SIP Registrar Serverů potřeba registrační údaje uživatele. Protože se tato bakalářská práce zaměřuje na spojení typu bod-bod, jsou tyto údaje ve spolupráci se SIP knihovnou pevně zaneseny do programového kódu zařízení. Hlavním a nejdůležitějším údajem, který v této fázi uživatel zadává, je adresa volaného zařízení. V případě, že všechno proběhne v pořádku, přejde program plynule do bloku „Diagram průběhu hovorů“, jehož vnitřní struktura bude dále rozebrána (Obr. 4.2).



Obr. 4.2: Vývojový diagram bloku „Diagram průběhu hovorů“

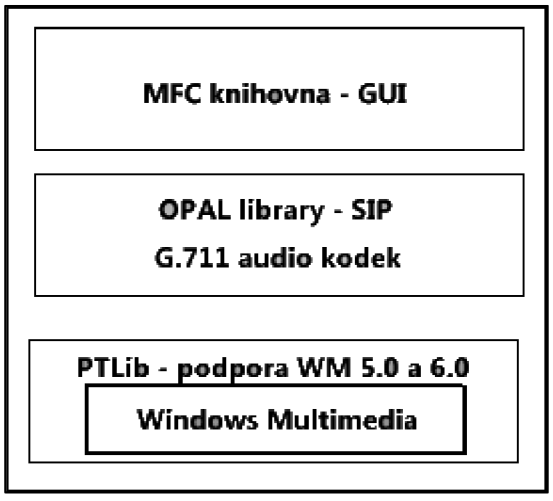
V tomto bloku je znázorněn model zpracování příchozích a odchozích hovorů. V prvním bodě dochází ke kontrole příchozího hovoru. V případě kladného vyhodnocení je spuštěn proces, při kterém jsou vyhodnoceny příchozí data pro přijetí komunikace, a uživatel je

upozorňován na tuto skutečnost. V této fázi má možnost odmítnutí, či přijmutí hovoru. Dále už komunikace probíhá podobně jako při klasickém telefonním hovoru. Běh se následně vrací opět do počáteční podmínky tohoto vývojového diagramu. Při kontrole příchozího hovoru je ještě možnost vyhodnotit podmínku záporně. Za tohoto předpokladu je vyhodnocována možnost vytočení jiného účastníka a jsou shromážděna data k provedení tohoto požadavku. Při přechodu do fáze vytáčení a hovoru je předpokládáno s obdobným zakončením jako při průchodu druhou větví pro přijetí hovoru.

Navržené vývojové diagramy byly vytvořeny pro ujasnění a usnadnění vytváření programu. Je zřejmé, že se diagramy v některých částech mohou cyklicky zastavit. Toto je způsobeno lineární strukturou diagramů, kdy nejsou zahrnuty veškeré proměnné a nejsou dořešeny výstupy z programu. Nejedná se tedy o přísnou definici fungování programu, ale o obecný rastr struktury UA.

4.3 Struktura realizované aplikace

Struktura výsledného programu je již z velké části předurčena zvolenou SIP knihovnou. Jak již bylo řečeno, jako programovací prostředí bylo použito Visual Studio 2005 a programovací jazyk C++. Celý návrh je založený na grafické uživatelské rozhranní MFC, OPAL library s implementovanou SIP knihovnou a RTP knihovnou a PTLib podporující funkce Windows Mobile mimo jiné s možnostmi příjmu a zpracovávání zvukového přenosu (Obr. 4.3).

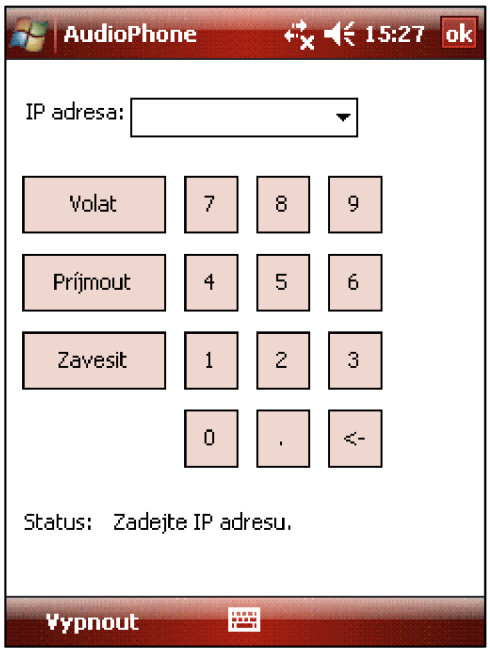


Obr. 4.3: Schéma programu a využívání

4.3.1 Uživatelské rozhranní MFC

Knihovna MFC je jedna z nejrozšířenějších knihoven pro vytváření grafického rozhraní pro operační systém Windows, která v sobě kolaboruje součástí Windows API do ucelených

C++ tříd. Protože se jedná o balík vyvíjený přímo společností Microsoft, jeho kompatibilitu rozšířili prakticky na všechny své operační systémy, tedy i na Windows Mobile. V projektu je toto GUI využito ke kompletní interakci mezi uživatelem a dalšími třídami. V programu je definována v souborech MOPALppc.rc a MobileOPALppc.rc2. Zahrnují v sobě informace o velikosti a umístění ovládacích prvků, události při práci s programem a také funkce hardwarových tlačítek, které využívají přenosná zařízení. Ukázku grafického uživatelského rozhraní viz Obr. 4.4.



Obr. 4.4: Ukázka grafického rozhraní programu ve Windows Mobile 6.0

4.3.2 Knihovna OPAL

V knihovně OPAL library byl z několika dostupných možností kódování přenášeného zvuku vybrán kodek G.711. Tento je prozatím nejrozšířenějším kodekem využívaným při VoIP telefonii, čímž je zajištěna vysoká kompatibilita při spolupráci s dalšími SIP UA. Tento kodek je umístěn ve třídě `g711codec`. Hlavní třída pro správu SIP je `handler`. Ve stejném umístění je třída `sdp`. Další důležitou součástí pro program je třída `rtp`.

4.3.3 Knihovna PTLib

Tato knihovna obsahuje třídy s metodami, které jsou potřeba pro funkčnost knihovny OPAL. Obsahuje poměrně obsáhlý seznam tříd, z nichž byla použita pouze část potřebná pro tento projekt. Jedná se o přístup a ovládání jednotlivých hardwarových prvků, jako jsou zvuk, síťové komponenty, vstupní a výstupní zařízení.

4.3.4 Třídy MODIg a MOPAL

Třída `MOGIg` je hlavní třídou programu. Přistupuje ke knihovně OPAL pomocí tříd `opalplugin` a `opalpluginmgr`. Kodeků z této knihovny využívá pomocí třídy `transcoders`. Z knihovny PTLib prostřednictvím třídy `sound` používá metody k nastavení zvukového výstupu. Počátečními jsou metody `OnInitDialog` a `InitialiseOPAL`. Zavádí se tady síťové rozhraní, o což se stará metoda `GetNetworkInterfaces`. Pomocí `ShowWindow` dochází k úpravě velikosti podle vlastností zařízení, na kterém je program spouštěný. Probíhá také zavádění informací k možnosti vytvořit a přijmout hovor. Třída `MODIg` obsahuje metody, které obsluhují události nastalé v uživatelském rozhraní – `OnBnClickedBtn1`, `OnBnClickedBtn2`, `OnBnClickedBtn3`, `OnBnClickedBtn4`, atd. Důležitými metodami pro řízení hovoru jsou `OnBnClickedBtnanswer`, `OnBnClickedBtndrop` a `OnBnClickedBtncall`, které zpracovávají požadavky uživatele volat, příjmout a odmítnout hovor.

4.4 Instalace programu do zařízení

Při kompilaci projektu byly složce Release vytvářeny ve všechny soubory, které jsou potřeba pro spouštění programu v emulátoru systému Windows Mobile. Soubory v dané složce je možno nahrát do mobilního zařízení a spouštět program z umístění, kde byl uložen. Zajímavějším a rozšířenějším způsobem je instalace programu do prostředí Windows Mobile. K tomuto účelu je zapotřebí soubor CAB, který je instalačním programem pro přenosná zařízení. Vytvořen byl ze souborů ve složce Release nástrojem pro tvorbu Smart device CAB projektů ve Visual Studiu. Zároveň se soubory ve složce byly do instalačního balíčku CAB přidány knihovny atl80, MFC80U a msvcr80. Knihovny byly dodány společně se SIP knihovnou pro práci v přenosných zařízeních. Takto zkompilovaný balíček je možné nainstalovat do zařízení běžným způsobem.

5 TEST PROGRAMU

Problematika testování výsledného softwarového řešení je poměrně rozsáhlé téma, které však bývá při vývoji programů opomíjené [3]. Poněvadž se jedná o velice důležitou součást celého projektu, bude jí věnována samostatná kapitola.

Při samotném vývoji programu byla snaha o co nejjednodušší eliminaci neošetřených výjimek, a to hlavně intuitivním ovládáním programu, kdy se uživateli neposkytuje příliš možností pro zadání chybného požadavku. A když taková situace přece jen nastane, není tento příkaz zpracován. Výsledkem této filozofie je jednoduchost a přehlednost programu. Ostatní chyby související se spouštěním, během a ukončením programu při využití všech implementovaných knihoven byly postupně odstraňovány při programování UA.

Samotné testování probíhalo na základě Black Box metody. Tato metoda je založena na předpokladu neznalosti vnitřní struktury programu, což bude při tomto způsobu testování simulováno. Vstupní hodnoty tedy budou postupně měněny a na jejich základě se bude přepokládat určitý scénář chování programu. Na konci budou hodnoty porovnávány se skutečným výsledkem.

5.1 Spojení mezi UA v systému WM6.0 a UA v systému WinXP

Tento test byl zařazen z důvodu snadnější kontroly probíhajícího přenosu mezi UA. Jedná se o spojení mezi navrženým klientem pro Windows Mobile s klientem Juphonne SIP Phonne. Tuto demoverzi nabízí volně přístupnou na svých internetových stránkách společnost Juphonne. Tento UA je založen na SIP knihovně v rámci developerského vývojového nástroje IPPhone Dev, který je také dílem této společnosti [5]. Výhodou tohoto UA a hlavním důvodem pro jeho výběr bylo jeho bezproblémové fungování i bez Registrar Serveru.

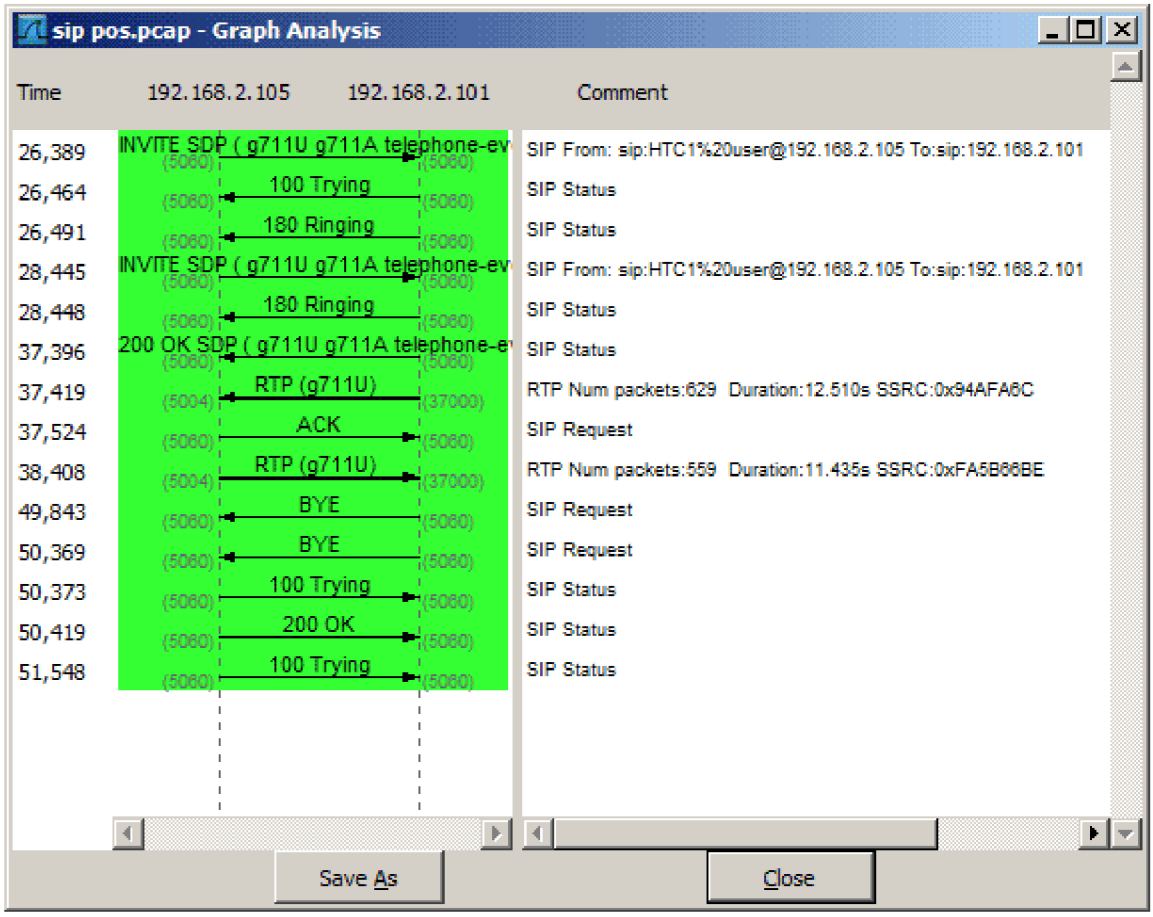
Předpokladem tohoto testu je spolupráce SIP signalizace u odlišných aplikací pro různé operační prostředí a následný přenos hovoru kódovaného stejným audio kodekem G.711. Komunikace bude testována v obou směrech kontrolou správné funkce signalizace a navázáním hlasového přenosu. K záznamu všech potřebných zpráv během komunikace byl použit program WireShark.

V rámci tohoto testování bylo provedeno hned několik variant možností probíhajícího spojení. Ke spojení byla použita bezdrátová síť, do které byla obě zařízení připojena lokálně. Pro přehlednost bude UA nainstalovaný na zařízení HTC Touch HD s IP 192.168.2.105 označován jako Mobil. Ve Windows XP bude UA s IP 192.168.2.101 dále označován jako XP. Jednotlivé testy budou rozebrány dál (Tab. 5.1).

Tab. 5.1: Testy provedené pro UA ve WM6.0 a ve Windows XP

číslo testu	první předpoklad	druhý předpoklad
1	Mobil vytočí IP adresu XP.	XP přijme hovor.
2	Mobil vytočí IP adresu XP.	XP odmítne hovor.
3	XP vytočí IP adresu Mobilu.	Mobil přijme hovor.
4	XP vytočí IP adresu Mobilu.	Mobil odmítne hovor.
5	Mobil vytočí IP adresu XP během probíhající relace.	Mobil přijme hovor až po ukončení předchozího.
6	Mobil vytočí IP adresu XP.	Mobil ukončí spojení před obdržením odpovědi.
7	XP vytočí IP adresu Mobilu.	XP ukončí spojení před obdržením odpovědi.

Test číslo 1 proběhl podle předpokládání, kdy po vytočení IP adresy XP upozorňoval na příchozí hovor. Během přenosu došlo k odeslání zprávy INVITE dvakrát (Obr. 5.1). Toto bylo způsobeno pravděpodobně bránou firewall počítače, protože při dalších pokusech o spojení byla zpráva odesílána pouze jedenkrát. Následně po přijetí hovoru nastal multimediální přenos a obě strany byly schopny spolu komunikovat. Nakonec došlo k požadavku o ukončení relace ze strany XP a spojení bylo korektně ukončeno. V následujícím textu je uveden přesný obsah první zprávy INVITE včetně SDP zprávy, která byla použita v této relaci.



Obr. 5.1: Výsledek testu – navázání spojení.

INVITE sip:192.168.2.101 SIP/2.0

Date: Tue, 26 May 2009 21:26:08 GMT

CSeq: 1 INVITE

Via: SIP/2.0/UDP 192.168.2.105:5060;branch=z9hG4bK0086497a-85fe-1810-94a0;rport

User-Agent: AudioPhone/1.0

From: "HTC1 user" <sip:HTC1%20user@192.168.2.105>;tag=0086497a-85fe-1810-949

Call-ID: 0086497a-85fe-1810-949b-001841d66c6d@HTC1

Organization: Philip Regueyra

To: <sip:192.168.2.101>

Contact: <sip:HTC1%20user@192.168.2.105>

Allow: INVITE,ACK,OPTIONS,BYE,CANCEL,SUBSCRIBE,NOTIFY,REFER,MESSAGE

Content-Type: application/sdp

Content-Length: 299

Max-Forwards: 70

v=0

o=- 1243373168 1243373168 IN IP4 192.168.2.105

s=Opal SIP Session

c=IN IP4 192.168.2.105

t=0 0

m=audio 5004 RTP/AVP 0 8 101 102

a=sendrecv

a=rtpmap:0 PCMU/8000/1

a=rtpmap:8 PCMA/8000/1

a=rtpmap:101 telephone-event/8000

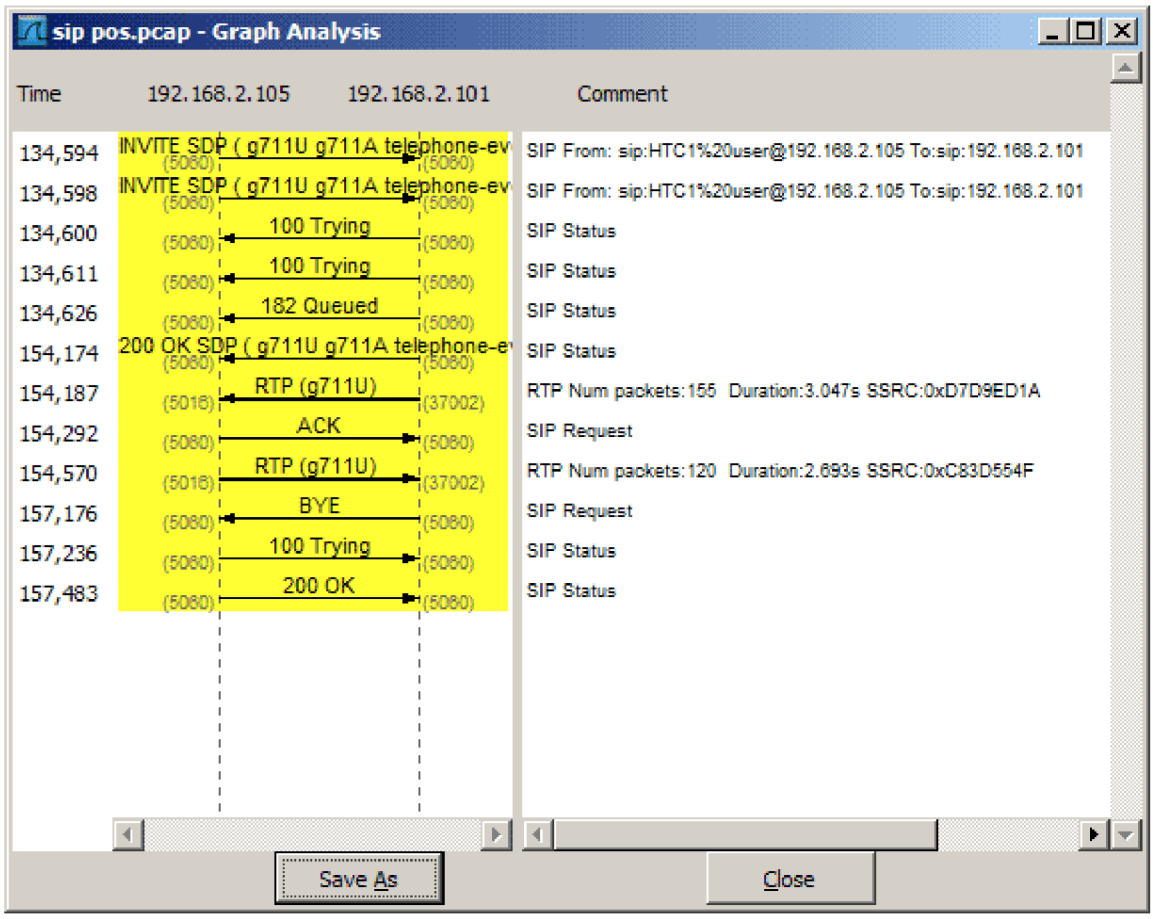
a=fmtp:101 0-16,32,36

a=rtpmap:102 NSE/8000

a=fmtp:102 192-193

Test číslo 2 dopadl podle předpokládaného scénáře. Při pokusu o navázání spojení UA Mobil byla ze strany XP odeslána zpráva o zamítnutí spojení a celá relace byla ukončena.

Testy číslo 3 a 4 probíhaly obdobně jako předchozí měření, pouze s rozdílem, že si Mobil a XP vyměnily roli UA klienta a UA serveru.

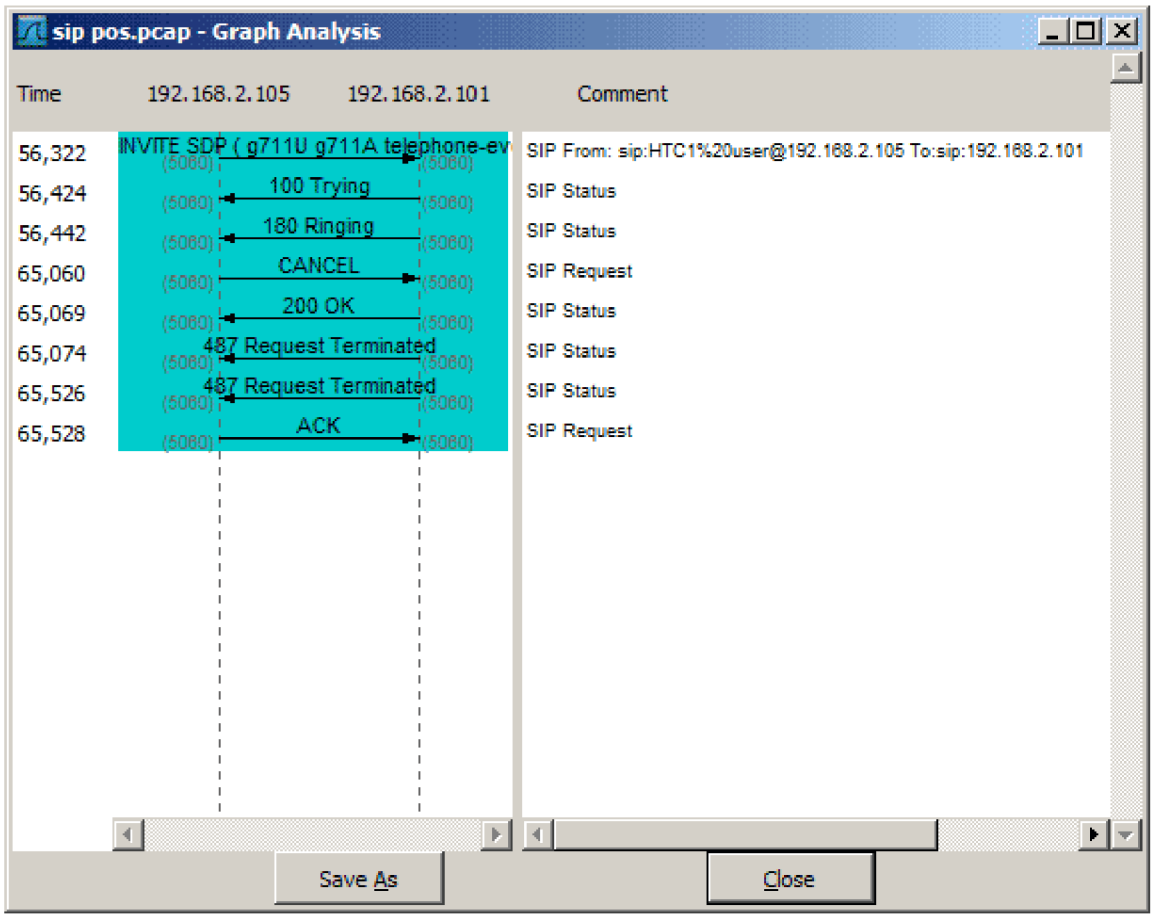


Obr. 5.2: Výsledek testu – přidržení hovoru.

U dalšího testu jsme obdrželi odpověď 182 Queued, kdy UA XP zařadil hovor do fronty, protože ve stejnou dobu se účastnil jiného hovoru (Obr. 5.2). Po skončení předchozího hovoru se XP rozhodl přijmout přidržený hovor, a to potvrzovací zprávou 200 OK. Tento případ byl pátým prováděným testováním (Tab. 5.1).

Poslední dvě testování byla z hlediska testovaného scénáře opět stejná, znovu došlo k prohození funkce serveru za klienta. UA Mobil po vytočení a čekání na odpověď hovor zavěsil, čímž ukončil relaci (Obr. 5.3). Reakcí na tuto skutečnost byla zpráva 487 Request Terminated odeslaná UA XP.

Všechny uvedené testy byly i při jejich opakování úspěšné. Vždy docházelo ke správnému dekódování SIP zpráv a k adekvátním odpovědím na ně. Někdy docházelo ke dvojnásobnému odeslání stejné SIP zprávy, což je nejspíše způsobeno nastavením brány firewall. Přesto hovory dle zadaných hodnot probíhaly přesně podle předpokladu.



Obr. 5.3: Výsledek testu – přerušení hovoru.

5.2 Komunikace mezi dvěma UA nainstalovanými na WM6.0

Program byl odzkoušen na dvou smartphonech. Jedním byl HTC Touch HD a druhým HTC Touch Cruiser. Zařízení byly připojeny do lokání sítě prostřednictvím bezdrátového routeru.

Došlo k provedení obdobných testů jako v části 5.1. Monitoring sítě nebyl v tomto případě možný, protože program pro sledování paketů dle našich potřeb pro WM neexistuje. Proto nebyly sledovány vyměněné zprávy, ale testy byly soustředěny na otázku funkčnosti UA.

U jednotlivě prováděných testů se předpokládaly obdobné výsledky jako v testování na pevné stanci a mobilním zařízení. Většina testů probíhala na základě vstupních hodnot přesně podle předvídaných scénářů. Hovor byl přijat nebo zamítnut podle požadované odpovědi jednotlivých stran. Odlišnost oproti předchozím výsledkům byla u testu číslo 5, kdy při probíhající hovoru UA nereagoval na žádost o nově provedený hovor. Tato skutečnost však není třeba považovat za chybu programu. Jedná spíše o vlastnost SIP knihovny, která neobslouží další příchozí hovor.

6 ZÁVĚR

Cílem práce bylo navrhnout a realizovat funkčního SIP klienta, který bude schopný uskutečnit spojení typu bod-bod. Prvotní myšlenkou bylo navrhnout aplikaci v jazyce C# pro prostředí Windows a následně provést jeho převod do systému Windows Mobile. Při realizaci programu se tato možnost ukázala jako nevhodná. Přepracování knihoven nSIP a nSDP pro zajištění jejich kompatibility v mobilních zařízeních by bylo časově velice náročné.

Z tohoto důvodu byly vybrány jiné knihovny, konkrétně PTLib a OPAL library, které jsou plně kompatibilní se systémem Windows Mobile. Následný vývoj SIP klienta byl prováděn přímo pro Windows Mobile. Bylo zvoleno vývojové prostředí Visual Studio 2005 rozšířené o vývojové balíčky Windows Mobile SDK 6. Výhodou vybraných jmenovaných knihoven PTLib a OPAL library je jejich komplexnost. Neobsahují pouze signalizační protokol SIP, ale celou škálu dalších knihoven zajišťujících správu síťových rozhraní, správu zvukového zařízení a multimediální protokol RTP s možností výběru několika kodeků.

Testy prováděné mezi pevnou stanicí a zařízením se systémem Windows Mobile byly zaměřené na vyzkoušení nejčastějších možných scénářů při telefonování. Obdobné testy byly prováděné mezi dvěma mobilními zařízeními. Během prováděných testů se neprojevily chyby ovlivňující funkčnost aplikace, čímž byly splněny hlavní požadavky kladené na realizovaný program.

Jak je z práce zřejmé, cíle vytyčené v zadání bakalářské práce se podařilo splnit. Nelze však tvrdit, že vývoj aplikace je zcela ukončen. Bylo by možné její funkce rozšířit o možnost registrace k Registrar Serveru. Dalšími vhodnými náměty pro rozšíření aplikace by mohly být podpora výběru z více audio kodeků, popřípadě i podpora výběru z video kodeků. Tyto myšlenky nejsou nezbytně nutné k využívání SIP UA, ale jejich možná realizace by zvýšila atraktivitu programu, který by mohl být rozšířen i mezi další eventuální uživatele.

LITERATURA

[1] JOHNSTON, Alan B. *SIP: Understending the Session Initiation Protocol*. Cover design by Lisa Johnson. 2nd edition. London: Artech House Boston, 2004. xxiii, 283 s. ISBN 1-58053-655-7.

[2] KLAUS, Darilion. *VoIP - SIP and RTP stacks, softphones, user agents, STUN - a comparison* [online]. 2004, 2004/05 [cit. 2008-12-11]. Text dostupný v angličtině. Dostupný z WWW: <http://www.pernau.at/kd/voip/bookmarks-sip-stacks.html>.

[3] NADRCHAL, Štěpán P. Je testování software věda?. *PDQM* [online]. 2005 [cit. 2009-05-19]. Dostupný z WWW: <http://www.pdqm.cz/Publications/TestVeda.pdf>.

[4] *NSIP / nSDP Project Page* [online]. August Sodora III, c2008 [cit. 2008-12-13]. Text v angličtině. Dostupný z WWW: <http://nsip.sourceforge.net/index.php>.

[5] *Open SIP Phone* [online]. Juphone, [2008] [cit. 2009-05-19]. Text v angličtině. Dostupný z WWW: <http://www.juphoon.com/ipphone/index.htm>.

[6] SAIDL, Martin. *Protokol SIP* [online]. 2002-02-21, 2002-02-21 [cit. 2008-12-07]. Dostupný z WWW: <http://saidl.tone.cz/text/html/VoIP/node22.html>. [6]

[7] *Receiving RTP Packets* [online]. Microsoft Corporation, 2008 [cit. 2008-12-12]. Text dostupný v angličtině. Dostupný z WWW: <http://msdn.microsoft.com/en-us/library/cc245393.aspx>.

[8] ROBINSON, Simon, et al. *C#: programujeme profesionálně*. Odpovědný redaktor Ivo Magera; překlad Borgan Kiszka. 1. vyd. Brno: Computer Press, 2003. xxx, 1130 s. ISBN 80-251-0085-5.

[9] *Session Initiation Protocol* [online]. Wikipedia: The Free encyklopidia, [2003], last modified on 5 December 2008 [cit. 2008-12-05]. Text dostupný v angličtině. Dostupný z WWW: <http://en.wikipedia.org/wiki/Session_Initiation_Protocol>.

[10] *Session Initiation Protocol* [online]. Wikipedia: La enciklopedia libre, [2005], modificada por última vez el 12 dic 2008 [cit. 2008-12-13]. Text dostupný ve španělštině. Dostupný z WWW: <http://es.wikipedia.org/wiki/Session_Initiation_Protocol>.

[11] SINNREICH, Henry, JOHNSTON, Alan B. *Internet Communications Using SIP: Delivering VoIP and Multimedia Services with Session Initiation Protocol*. 2nd edition. Indianapolis : Wiley Publishing, Inc., 2006. xxix, 377 s. ISBN 978-0-471-77657-4.

[12] *SIP, H.323 and IAX Audio Conference Server details* [online]. RSDevs.com, c2005-2009 [cit. 2009-04-30]. Text v angličtině. Dostupný z WWW: <http://www.rsdevs.com/mcu.shtml>.

[13] VIRIUS, Miroslav. *Jazyky C a C++ : kompletní kapesní průvodce programátora*. 1. vyd. Praha : Grada, 2006. 518 s. ISBN 80-247-1494-9.

[14] *Windows Mobile 6* [online]. Microsoft Corporation., c2009 [cit. 2009-03-11]. Text v češtině. Dostupný z WWW: <http://msdn.microsoft.com/en-gb/library/bb158486.aspx>.

SEZNAM ZKRATEK, VELIČIN A SYMBOLŮ

API Application Programming Interface

ASCII American Standard Code for Information Interchange

CAB Cabinet file format

GPL GNU General Public License

GUI Graphical User Interface

HTTP Hypertext Transfer Protocol

IM Instant Message

QoS Quality of Service

MFC Microsoft Foundation Classes

MPL Mozilla Public License

RFC Request For Comments

RTCP RTP Control Protocol

RTP Real-time Transport Protocol

SDK Software Development Kit

SDP Session Description Protocol

SIP Session Initiation Protocol

SMTP Simple Mail Transfer Protocol

TCP Transmission Control Protocol

UA User Agent

UDP User Datagram Protocol

UMA Unlicensed Mobile Access

URI Uniform Resource Identifier

VoIP Voice over IP

Win Windows

WM Windows Mobile

SEZNAM PŘÍLOH

A Obsah CD **44**

A OBSAH CD

- Elektronická verze bakalářské práce – bakalářská práce.pdf
- Knihovna PTLib umístěná v komprimovaném souboru ptil.zip.
- Knihovna OPAL library umístěná v komprimovaném souboru opal.zip.
- Nástroj na tvorbu instalačního balíčku v komprimovaném souboru Distri.zip.
- Instalační balíček aplikace AudioPhone.cab
- Zdrojový kód aplikace ve složce AudioPhone