

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informatiky a kvantitativních metod

Mobilní aplikace pro podporu testování MMS centra
Bakalářská práce

Autor: **Dominik Plašil**
Studijní obor: Aplikovaná informatika
Vedoucí práce: Ing. Pavel Kříž, Ph.D.
Odborný konzultant: Ing. Pavel Dejmek

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 12. dubna 2018

Dominik Plašil

Poděkování:

Děkuji vedoucímu bakalářské práce Ing. Pavlu Kříži Ph.D. za metodické vedení práce, cenné připomínky a rady během tvorby této práce.

Anotace

Práce se zaměřuje na popis aplikace vyvíjené pro společnost O2 Czech Republic a.s. a případně další uživatele používající mobilní zařízení s operačním systémem Android.

V úvodu práce jsou popsány SMS, EMS a MMS zprávy obecně včetně jejich historie. Následuje představení technologie MMS zpráv. To zahrnuje popis, jak jsou MMS zprávy vytvářeny a odesílány na server mobilního operátora, jak je server zpracuje a pošle příjemci, vč. výměny zpráv s informacemi o doručení a přečtení. Tato část také zahrnuje popis standardizace a specifikace pro MMS zprávy, jak fungují MMS centra a co musí být v mobilním zařízení nastaveno, aby bylo možné odeslat zprávu. Následuje část s popisem Android API pro práci se MMS zprávami. Druhou část práce tvoří analýza a vývoj vlastní aplikace podle zadání doplněná rešerší několika již existujících řešení. V závěru práce jsou uvedeny některé problémy vzniklé při vývoji aplikace a celkové shrnutí.

Annotation

Title: Phone Application Supporting Testing of MMS Center

This Bachelor Thesis focuses on the description of application developed for company O2 Czech Republic a.s. and many other users who use phone devices with Android operation system.

The first part describes SMS, EMS and MMS messages in general including their history. Followed by description of MMS messages technology. This includes how MMS messages are created and sent to the mobile operator server, how server handles them and finally how delivery and read reports work. Moreover, this part contains few lines about MMS standardization and specifications, how MMS centers work and what needs to be setup on the phone device to be able to send messages.

Next part focuses on description how Android API for MMS messages works.

The second half of the work describes the analysis and development of own application based on assignment including few lines about existing solutions. And the end is devoted to description of few problems which occurred while development and to final summary of the work.

Obsah

1	Úvod.....	1
2	Cíl práce.....	2
3	SMS, MMS a související technologie.....	3
3.1	SMS zprávy.....	3
3.2	EMS zprávy	3
3.3	MMS zprávy	4
4	Standardizace	5
4.1	Standardizace MMS	5
4.1.1	3GPP.....	5
4.1.2	3GPP2	6
4.1.3	W3C.....	6
4.1.4	OMA.....	6
4.2	Základní architektura sítě.....	7
4.2.1	MMS Centra.....	8
4.3	MMS rozhraní.....	9
4.3.1	Rozhraní MM1.....	9
4.3.2	Rozhraní MM4.....	9
4.4	PDU balíčky.....	10
4.5	SIM Karta	10
4.5.1	Údaje uložené na SIM kartě	11
4.6	APN	12
5	Android API pro práci s SMS a MMS	13
5.1	MMS PDU v Android API.....	13
5.1.1	Pomocné třídy pro PDU balíčky	14
5.2	Scénář práce s PDU třídami.....	16

5.2.1	Message ID.....	17
5.2.2	Odeslání zprávy	17
5.2.3	Přijetí zprávy	17
5.2.4	Okamžité nebo odložené stažení	18
5.2.5	Zprávy o doručení a přečtení	19
5.3	Nastavení aplikace pro práci se zprávami	20
5.4	Rozdíly v MMS API na různých verzích Android.....	21
5.4.1	APN a Subscription ID.....	21
5.4.2	API pro odeslání MMS.....	22
5.4.3	API pro přijetí a stažení MMS.....	22
5.4.4	Práce s RR a DR.....	23
5.5	Možnosti ukládání zpráv	24
6	Vývoj mobilní aplikace	25
6.1	Zadání a seznam požadavků	25
6.2	Rešerše existujících řešení.....	25
6.2.1	Knihovna android-smsmms	25
6.2.2	WhisperSystems/Signal-Android.....	26
6.2.3	Výchozí aplikace na Android a podobné aplikace na Google Play	27
7	Analýza a návrh aplikace.....	29
7.1	Aplikace MMS Params	29
7.2	Analýza požadavků	29
7.2.1	Diagram případů užití.....	30
7.3	Profily nastavení	31
7.4	MMS šablony	33
7.5	Zobrazení PDU balíčků zpráv	34
7.5.1	Výpočet velikosti zprávy.....	36

7.6	Job Manager.....	37
7.7	Firestore.....	37
7.7.1	Logování chyb na Firestore.....	38
7.7.2	Testy aplikace na Firestore.....	38
7.8	Rozšiřitelnost aplikace.....	38
7.9	Problémy.....	39
7.9.1	Chybějící parametry v PDU třídách.....	39
7.9.2	Velikost PDU balíčků.....	40
7.9.3	Nastavení APN v aplikaci.....	40
7.10	Jak aplikaci získat.....	40
8	Závěr.....	41

Seznam obrázků

Obrázek 1 Schéma MMS a dalších sítí [12]	7
Obrázek 2 Základní architektura MMS sítě [13]	8
Obrázek 3 MMS Centrum a jeho MM rozhraní [12]	9
Obrázek 4 Schéma komunikace pomocí MM4 rozhraní [12]	10
Obrázek 5 Sekvence odesílání a přijímání PDU balíčků [12]	16
Obrázek 6 Okamžité stažení MMS zprávy [13]	18
Obrázek 7 Odložené stažení MMS zprávy [13]	19
Obrázek 8 Nastavení ve výchozí aplikaci	27
Obrázek 9 Nastavení MMS v aplikaci Chomp SMS	27
Obrázek 10 Diagram případů užití MMS Params	30
Obrázek 11 Třída MmsProfile	31
Obrázek 12 Detail nastavení profilu	32
Obrázek 13 Detail nastavení profilu	32
Obrázek 14 Výběr času expirace zprávy	33
Obrázek 15 Detail nastavení MMS šablony	34
Obrázek 16 Detail PDU SendReq	35
Obrázek 17 Detail PduBody	35
Obrázek 18 Detail PDU pro přijatou zprávu	36

Seznam ukázek kódů

Ukázka kódu 1 Vygenerovaný SMIL	16
Ukázka kódu 2 Zobrazení dialogu pro přepnutí výchozí aplikace	20
Ukázka kódu 3 Získání seznamu aktivních SIM	21
Ukázka kódu 4 Metoda onReceive	23
Ukázka kódu 5 Zpracování přijatého PDU balíčku	23
Ukázka kódu 6 Načtení odeslaných SMS	24
Ukázka kódu 7 Uložení odchozí MMS zprávy	24
Ukázka kódu 8 Nastavení parametrů zprávy v knihovně android-smsmms	26
Ukázka kódu 9 Metoda pro výpočet velikosti zprávy	36

Použité zkratky a pojmy

SMS	<i>Short Message Service</i> služba pro krátké textové zprávy
EMS	<i>Enhanced Messaging Service</i> služba pro textové zprávy rozšířené o mediální obsah
MMS	<i>Multimedia Messaging Service</i> služba pro multimediální zprávy
API	<i>Application Programming Interface</i> soubor tříd, funkcí a protokolů, který umožňuje vytvářet aplikace, které mají přístup k datům nebo funkcím operačního systému, jiné aplikace nebo služby
GSM	<i>Global System for Mobile Communications</i> služba pro bezdrátovou komunikaci mobilních zařízení
GPRS	<i>General Packet Radio Service</i> služba pro bezdrátovou komunikaci používající pakety
LTE	<i>Long-Term Evolution</i> standard pro vysokorychlostní bezdrátovou komunikaci mobilních zařízení
APN	<i>Access Point Name</i> brána mezi GSM, GPRS, 3G, 4G, dalšími mobilními sítěmi a počítačovou sítí nebo sítí internet
SIM	<i>Subscriber Identity Module</i> integrováný čip, který uchovává autentizační hodnoty uživatele zařízení
PDU	<i>Protocol Data Unit</i> datová jednotka přenášená mezi jednotlivými zařízeními v mobilní nebo počítačové síti
MMSE	<i>Multimedia Messaging Service Environment</i> skupina prvků pro specifickou MMS síť spravovaná jedním subjektem

1 Úvod

Mobilní aplikace pro přijímání a odesílání SMS a MMS zpráv, kterou jsou nainstalovány od výrobce telefonu většinou neumožňují nastavit mnoho parametrů zpráv. Často je to pouze telefonní číslo příjemce, text zprávy a jeden nebo více obrázků nebo souborů. Výchozí APN mobilního operátora je také většinou nastaveno automaticky s vloženou SIM kartou. Toto nastavení a funkce jsou pro většinu uživatelů mobilních zařízení naprosto dostatečné a uživatelé často nepotřebují vůbec nic měnit nebo nastavovat. Nicméně pro účely testování MMS center, je to naprosto nedostatečné.

Tato práce popisuje stručnou historii zpráv na mobilních zařízeních od SMS zpráv, přes EMS až po MMS zprávy, které budou dále rozebrány do podrobná. Jsou popsány jejich vlastnosti, technologie přenosu a co vše je součástí komunikace pomocí MMS zpráv. Aplikace pracující s MMS zprávami musí dodržovat zásady ze specifikace 3GPP a OMA, aby bylo zajištěno správné fungování a kompatibilita s MMS centry mobilních operátorů. Jednou z těchto zásad je zpracování a komunikace pomocí PDU balíčků zpráv. V této práci jsou popsány tyto PDU balíčky obecně a následuje přehled, jak jsou tyto balíčky a potřebné nástroje s nimi spojené implementovány v operačním systému Android. Druhou část práce tvoří analýza a vývoj aplikace pro mobilní zařízení s operačním systémem Android dle konkrétního zadání od českého mobilního operátora O2 Czech Republic a.s.

2 Cíl práce

Cílem práce je popsat, jak fungují MMS zprávy na mobilních zařízeních a jaké technologie a specifikace se používají pro jejich odesílání, přijímání a zpracování. Dalším cílem je vytvořit mobilní aplikaci pro operační systém Android, která bude sloužit pro podporu testování MMS center a bude umožňovat nastavení různých parametrů pro odesílání a přijímání MMS zpráv. Aplikace bude umět MMS zprávy přijmout, získat z nich všechny parametry a informace a zobrazit je. Mimo jiné lze předdefinovat různé profily nastavení a mezi nimi rychle přepínat. To usnadní samotné testování. Lze také předdefinovat šablony MMS zpráv, které je možné používat opakovaně a testovat s nimi MMS centra.

3 SMS, MMS a související technologie

SMS a MMS zprávy jsou jednou z nejpoužívanějších a nejpoužívanějších komunikačních služeb, ve kterých se zprávy posílají elektronicky. Snížení poplatků za tyto služby vedlo k rapidnímu zvýšení využívání těchto služeb. Avšak v posledních letech tento trend stagnuje [1], kvůli vysoké dostupnosti mobilního internetu a nepřehledného množství dalších komunikačních nástrojů, které jsou většinou bez poplatků (Email, Facebook, WhatsApp).

3.1 SMS zprávy

SMS (Short Message Service) je základní služba umožňující výměnu krátkých textových zpráv mezi zařízeními. První textová zpráva byla poslána v roce 1992 a od té doby tato služba zaznamenala výrazný vzestup. The Mobile Data Association¹ zveřejnila, že v roce 2013 bylo jen ve Velké Británii posláno 20,5 miliardy těchto zpráv. [2] Původně byla standardizace SMS vytvářena ETSI (European Telecommunications Standards Institute), ale nyní je plně pod záštitou 3GPP. Dříve fungovala SMS v GSM síti, později byla ale převedena do jiných sítí (např. GPRS a CDMA).

Kvůli technologii SMS byla zavedena takzvaná SMS centra, která hrají klíčovou roli v architektuře služby SMS. Hlavní funkcí je přenos zpráv mezi jednotlivými uživateli a také store-and-forwarding (uložení a přesměrování), pokud není právě příjemce zprávy dostupný. SMS centrum může být také umístěno mimo síť samotnou a spravováno organizací třetí strany. V praxi je také běžné, že jeden operátor má více SMS center a teoreticky může být jedno SMS centrum použito pro více mobilních sítí, avšak to nastává pouze zřídka. [2]

3.2 EMS zprávy

Ačkoli SMS zprávy byly a jsou významným komerčním úspěchem, v roce 2000 začaly slábnout. Jako prevenci se několik výrobců mobilních zařízení a mobilních operátorů rozhodlo rozšířit služby o EMS (Enhanced Messaging Service). EMS rozšiřuje SMS o přenos tzv. „rich-media“ zpráv. Tyto zprávy mohou obsahovat černobílé obrázky nebo animace a monofonní melodie. Jednou z vlastností je, že tato

¹ <http://www.mda-mobiledata.org/>

média mohou být umístěna přímo do textu. Nevýhodou je, že se vkládaný element musí vždy vejít do jedné zprávy a nemůže být rozdělen do více zpráv. Nicméně i přes tuto limitaci byl rozvoj a používání velmi značný a vznikaly i zcela nové editory umožňující například posílat formátovaný text. Později vzniklo rozšíření Extended EMS. Tyto zprávy dokonce umožňovali barevné obrázky, vCard, MIDI melodie, a dokonce i vektorovou grafiku. [2]

EMS skutečně nikdy nedosáhla očekávané úrovně v rámci marketingového růstu, i přes to, že byly podporovány v řadě zařízení. EMS zprávy byly pouze mezikrok mezi SMS a MMS a nyní je již zřejmé, že po globálním rozšíření MMS a internetu tyto zprávy již nikdy neuspějí.

3.3 MMS zprávy

Službu MMS můžeme považovat za to nejlepší, co vzniklo z SMS a EMS. Služba pro posílání MMS zpráv byla na trh uvedena v březnu 2002 a ihned zaznamenala velký úspěch. Více než 100 mobilních operátorů ji ihned uvedlo do provozu ve vlastní síti. Původně MMS značně závisela na dostupnosti barevného displeje. Hlavní obsah tvořily fotografie a videa, které by na černobílém displeji neměly příliš velký význam. Byl to také základ pro komunikaci v mobilní síti pomocí paketů.

Dokonce už i raná implementace MMS umožňovala posílat multimediální zprávy nejen mezi mobilními uživateli, ale také do sítě internet. Základ MMS tedy leží někde mezi SMS založenou pouze na textu a emailem na internetu. MMS zprávy převzaly mnoho funkcí právě z emailu. Mimo jiné také zprávy o doručení a přečtení, třídy zpráv nebo nastavení priority zprávy. MMS může být jednoduše označena jako email mezi uživateli mobilních zařízení, kde „adresu“ tvoří telefonní číslo. [2]

4 Standardizace

Standardizace telekomunikačních technologií a přidružených služeb má klíčový význam pro rozvoj komunikačních systémů hlavně v prostředích, kde se pohybuje hned několik dodavatelů různých systému nebo zařízení. Několik subjektů, jako jsou mobilní operátoři, výrobci zařízení a vývojáři aplikací, spolupracuje, aby byly vytvořeny společné technické specifikace, které budou standardizovat všechna komerční řešení. Hlavním cílem je identifikování nejvhodnějších prvků pro danou problémovou oblast a jejich následné fungování a spolupráce.

4.1 Standardizace MMS

SMS, EMS i MMS zprávy jsou předmětem významných standardizačních aktivit. Na rozdíl od standardizace SMS a EMS, se standardizace MMS služeb stala velmi komplexním problémem. MMS je sofistikovaná služba multimediálních zpráv, která vyžadovala obrovskou snahu a spolupráci několika standardizačních subjektů pro vytvoření technické specifikace a uvedení celé služby na trh.

Bylo zapotřebí práce několika standardizačních subjektů, které společně vytvořily stabilní technickou specifikaci. Hlavní význam měly organizace 3GPP a tzv. WAP Forum. WAP Forum bylo roku 2002 spojeno s dalšími subjekty a vytvořila se OMA (Open Mobile Alliance). 3GPP převzalo vedení hlavně při definování vysokoúrovňových požadavků na službu, návrhu architektury sítě, vytvoření několika nízko úrovňových technických realizací a identifikování vhodných kodeků, formátů a streamovacích protokolů.

Mnoho vytvořených standardů těmito organizacemi vycházelo z již existujících technologií vyvinutých zejména konsorciem W3C. [2] Každý, kdo vytváří nějaký produkt založený na službách SMS, EMS nebo MMS, se musí řídit těmito specifikacemi, aby byla zajištěna spolehlivost a kompatibilita jednotlivých zařízení v systému.

4.1.1 3GPP

3GPP (3rd Generation Partnership Project) byl založen roku 1997 a spojuje sedm organizací, vytvářejících standardy pro telekomunikaci, též známých jako „Organization Partners“. Původním záměrem bylo vytvořit globálně použitelnou specifikaci pro mobilní systémy třetí generace záložnou na specifikaci GSM.

Postupně však byla působnost rozšířena na GSM (2G a 2.5G sítě), UMTS (3G standardy) a LTE (4G standardy včetně LTE Advanced a LTE Advanced Pro). [3] Specifikace vytvořené 3GPP jsou rozděleny na dva druhy: TS (Technické Specifikace) a TR (Technické reporty). TS definují GSM/UMTS standard. TR zkoumají proveditelnost pro nové funkce a služby a občas se z nich stanou technické specifikace. Jejich hlavním účelem je naplnit stále se měnící požadavky trhu, proto musí být pravidelně aktualizovány. [2]

Dalším důležitým cílem specifikací 3GPP je udělat systémy zpětně kompatibilní. To je důležité zejména proto, aby bylo zajištěno nepřetržité fungování všech zařízení a služeb, které se těmito specifikacemi řídí. [3]

4.1.2 3GPP2

3GPP2 (Third Generation Partnership Project 2) byl vytvořen v prosinci roku 1998 a je tvořen spojením a spoluprací několika telekomunikačních společností. Jeho úkolem je vytvořit specifikace pro průmyslové hráče a trhy v Severní Americe a Asii. Zaměřuje se zejména na vytvoření specifikací pro mobilní systémy a sítě třetí generace (3G). [4]

4.1.3 W3C

W3C (World Wide Web Consortium) je mezinárodním konsorciem založeným roku 1994. Úkolem konsorcia je dohlížet na vývoj internetových standardů a zajistit jejich interoperabilitu (schopnost různých systémů vzájemně spolupracovat). [5]

W3C se skládá z několika stovek komerčních společností a akademických organizací a je financováno sponzory a členy (poplatek za členství). Během své existence W3C vytvořilo mnoho standardů například pro XHTML, XML, CSS či DOM. W3C se také zabývá přístupností webu. [6] Konsorciu předsedá jeho zakladatel Tim Berners-Lee, tvůrce služby web a autor specifikací URL, HTTP a HTML. [7]

4.1.4 OMA

OMA (Open Mobile Alliance) je organizací s více než 220 členy, která vytváří otevřené standardy a specifikace pro průmysl mobilních telefonů a telekomunikační služby. OMA byla založena nejvýznamnějšími světovými mobilními operátory a výrobci a dodavateli zařízení a sítí. OMA je nezisková organizace, která vytváří specifikace pro několik spolupracujících služeb. Ty mají fungovat bez ohledu na

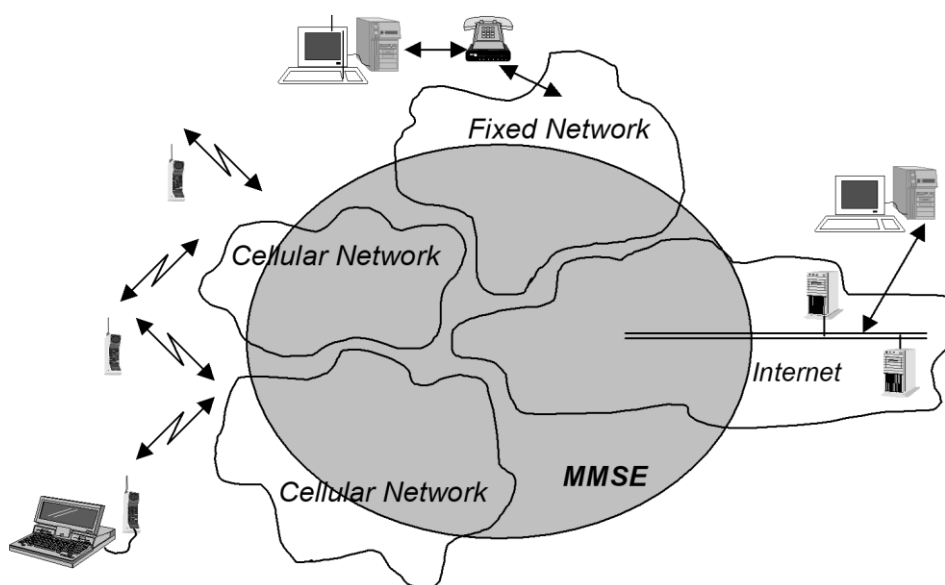
architekturu frameworku, operační systém, typ sítě, poskytovatele nebo geografickou lokalitu. [8]

Specifikace OMA jsou podporovány miliardami zařízení napříč několika bezdrátovými sítěmi, včetně všech tradičních mobilních operátorů. Dále jsou to sítě s podporou komunikace klient-klient a v neposlední řadě také zařízení IoT. [9]

Cílem OMA je vytvoření otevřených technických specifikací s vysokou kvalitou, založených hlavně na potřebách a požadavcích trhu. Cílem je možnost modularity, rozšiřitelnosti, konzistence a snížení nákladů a úsilí na implementaci. OMA má za cíl být katalyzátorem při konsolidaci standardních aktivit v rámci odvětví mobilních datových služeb [10]. OMA spolupracuje s dalšími standardizačními organizacemi, jakou jsou 3GPP nebo 3GPP2 a nemá za cíl s těmito organizacemi soupeřit. Spíše se snaží o vylepšení již existujících specifikací, aby byla vytvořena ještě větší interoperabilita služeb napříč všemi koncovými zařízeními. [11]

4.2 Základní architektura sítě

Obrázek 1 ukazuje schéma základní architektury MMS sítě. MMSE (Multimedia Messaging Service Environment) tvoří několik síťových prvků, které jsou kontrolovány a spravovány jedním subjektem. V MMSE lze kombinovat různé sítě a různé typy sítí pro zajištění komunikace mezi mobilními sítěmi (2G, 3G, LTE), sítí pevné linky a sítí internet. [12]

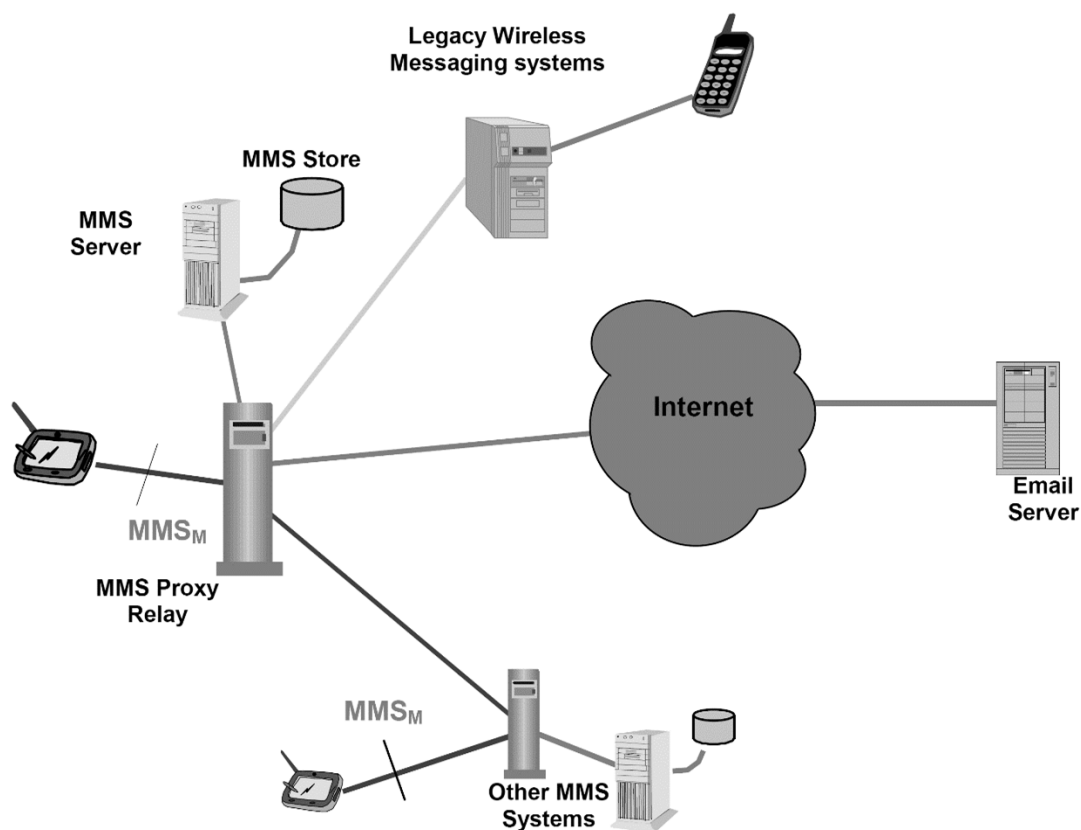


Obrázek 1 Schéma MMS a dalších sítí [12]

4.2.1 MMS Centra

MMS Centra (MMSC) jsou klíčovým prvkem v architektuře MMS sítě. Logicky je MMSC složeno z částí MMS Relay a MMS Server². MMS Relay je zodpovědné za přenos zpráv nejen v rámci jedné sítě, ale i do okolních sítí (např. do sítě jiného mobilního operátora). MMS Server má na starost hlavně ukládání odesílaných zpráv a informování příjemců o zprávách nových. Odsud si také příjemce zprávy stahuje. Server může mít vestavěné funkce pro překódování zpráv, správu uživatelů nebo ukládání uživatelských profilů. Tyto funkce mohou být i mimo MMSC samotné, například jako externí specializovaná komponenta. Aby mohl uživatel využívat služeb MMS centra, musí být zaregistrován [2].

Obrázek 2 pak podrobněji zobrazuje architekturu prostředí pro MMS zprávy. MMS Relay/Server je zodpovědný za ukládání a zpracovávání odchozích a příchozích zpráv a dokáže klienty projit s jinými sítěmi. [13]

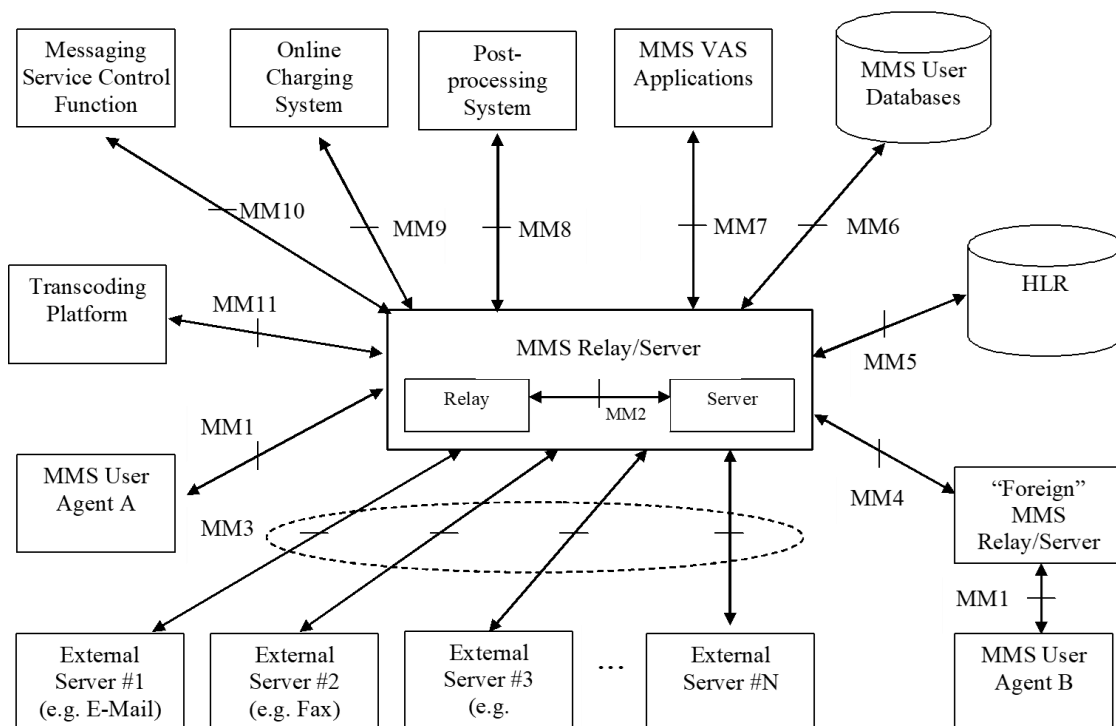


Obrázek 2 Základní architektura MMS sítě [13]

² Pojem MMS Centrum se někdy nahrazuje pojmem MMS Proxy/Relay nebo MMS Relay/Server [2]

4.3 MMS rozhraní

MMS Centra (resp. MMS Relay/Server) komunikují s okolím nebo mezi sebou pomocí určitých typů rozhraní. 3GPP ve své specifikaci označuje rozhraní jako MM a každý typ rozhraní má své vlastní označení. Jak ukazuje Obrázek 3, každé rozhraní je určeno pro specifickou službu, s kterou MMS centrum komunikuje [12]. Tato práce se hlavně zabývá rozhraním MM1, ale za zmínku stojí také rozhraní MM4.



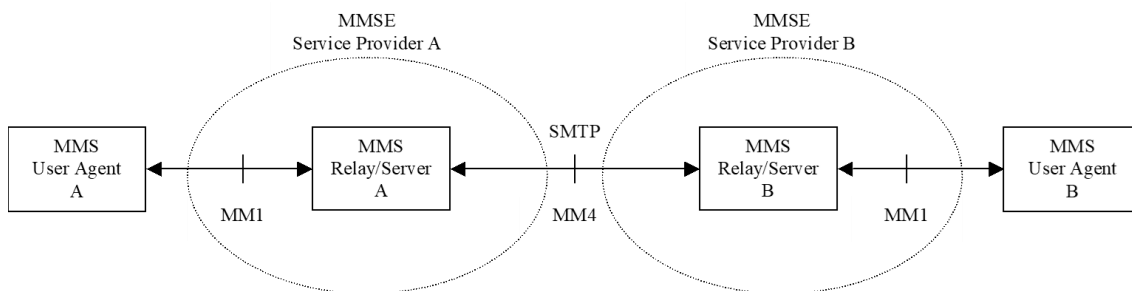
Obrázek 3 MMS Centrum a jeho MM rozhraní [12]

4.3.1 Rozhraní MM1

Rozhraní MM1 se používá pro odesílání multimediálních zpráv z mobilního zařízení do MMS centra, které následně pošle notifikaci příjemci o nové zprávě. Ten si může zprávu stáhnout a zobrazit. MMS centrum se dále stará o výměnu zpráv o doručení a přečtení mezi odesílatelem a příjemcem. [12]

4.3.2 Rozhraní MM4

Rozhraní MM4 spojuje dvě různá MMS Centra a je nezbytné pro výměnu multimediálních zpráv mezi různými mobilními operátory. Obrázek 4 také ukazuje, že přenos dat přes toto rozhraní je prováděn pomocí protokolu SMTP (Simple Mail Transfer Protocol). [2]



Obrázek 4 Schéma komunikace pomocí MM4 rozhraní [12]

4.4 PDU balíčky

PDU (Protocol Data Unit) je datová jednotka přenášená mezi jednotlivými zařízeními v mobilní nebo počítačové síti, které obsahuje informace o protokolu, adresy a vlastní datové informace [14]. PDU balíčky jsou nezbytnou součástí MMS zpráv a obsahují veškeré odesílané a přijímané informace. Ve specifikaci 3GPP je u každého MM rozhraní detailně popsáno, jaké PDU balíčky se mají využívat při komunikaci a jaké má být jejich pořadí (některé například slouží pro odeslání zprávy a jiné jako odpověď).

4.5 SIM Karta

The Subscriber Identity Module neboli SIM karta je základní a povinná součást mobilního telefonu. SIM karty mohou mít různou velikost³ a někdy je v určitých zařízeních není možné ani vyměnit. SIM karta je integrovaný čip, který uchovává autentizační hodnoty uživatele zařízení.

První SIM karty byly stejně velké jako běžné bankovní kreditní karty, ale postupem času se jejich velikost postupně snížila. Kontakty karty zůstaly stále stejné, proto je často možné kartu zmenšit pro moderní mobilní zařízení. SIM karty je možné různé přenášet mezi různými mobilními zařízeními a mobilní číslo přiřazené uživateli někdy nemusí být uloženo přímo na SIM kartě. [15]

³ <http://www.simcardwarehouse.com/sim-card-sizes>

4.5.1 Údaje uložené na SIM kartě

Každá SIM karta obsahuje několik identifikační a autentizačních údajů.

4.5.1.1 ICCID

Číslo ICCID (Integrated Circuit Card Identifier) je unikátní číslo SIM karty, které nemůže být za žádných okolností změněno a je i fyzicky vytisknuto na líc každé SIM karty. [16]

4.5.1.2 IMSI

IMSI (International Mobile Subscriber Identity) je unikátní číslo v GSM systému o maximální délce 15 znaků, které jednoznačně identifikuje účastníka mobilní sítě. Číslo je tvořeno třímístným číslem MCC, které identifikuje oblast nebo zemi a dvou nebo třímístným číslem MNC, které identifikuje mobilního operátora v dané oblasti nebo zemi. IMSI je dále tvořeno maximálně desetimístným číslem MSIN (Mobile Subscription Identification Number), které přiděluje mobilní operátor jako jednoznačnou identifikaci účastníka sítě. [16]

4.5.1.3 PIN

PIN (Personal Identification Number) je zastaralý termín pro CHV (Card Holder Verification Information). Opa pojmy označují 4–8místný kód, sloužící pro autentizaci uživatele k použití SIM karty. Poskytovatel SIM karty nastaví PIN kód na předvolenou hodnotu (obvykle 4místnou) a uživateli je doporučeno po prvním zapnutí telefonu s touto SIM kartou hodnotu změnit. PIN kód je dále rozdělen na PIN1 a PIN2 (resp. CHV1 a CHV2). PIN1 je nutný pro zapnutí telefonu a normální využívání SIM karty. Bez této autentizace není možné využívat služby SIM karty (kromě tísňového volání). PIN2 slouží pro správu některých vedlejších funkcí SIM karty. [16]

Pokud je PIN kód zadán třikrát špatně, telefon se zablokuje a pro jeho odemčení je nutné zadat PUK kód. Naopak pokud dojde k zablokování při zadávání PIN2, telefon lze i nadále používat normálně. Zablokují se pouze funkce vyžadující PIN2, dokud nedojde k jejich odblokování pomocí kódu PUK2. [17]

4.5.1.4 PUK

Kód PUK (PIN Unlock Key) je 8místný kód dodávaný společně se SIM kartou. Jak již bylo zmíněno, jeho úkolem je odblokovat SIM, pokud dojde k její zablokování při zadávání kódu PIN. Pokud je PUK kód zadán desetkrát chybně, SIM karta se zablokuje natrvalo a nelze ji již nikdy odblokovat – uživatel si musí koupit novou [16].

4.6 APN

APN (Access Point Name) je název brány mezi mobilními sítěmi (GSM, GPRS, 3G nebo 4G) a počítačovou sítí (nejčastěji sítí internet). Pro přenos dat musí mít telefon správně nastavené APN dle daného mobilního operátora. Každý telefon má několik výchozích nastavení APN a po vložení SIM karty je automaticky vybráno to nejvhodnější [18]. APN nastavení tvoří adresa MMSC, IP adresa Proxy a port. Dále může obsahovat např. uživatelské jméno a heslo pro připojení nebo typ autentizace. APN je zvoleno na základě identifikátoru IMSI načteného ze SIM karty [19]. Například Česká Republika má MMC číslo 230 a čeští operátoři mají přiřazena MNC čísla 01 pro T-Mobile, 02 pro O₂ a 03 pro Vodafone. Více informací a přehledný seznam všech kódů lze najít například na stránce www.mcc-mnc.com.

5 Android API pro práci s SMS a MMS

Operační systém Android má propracované a poměrně jednoduché API pro práci SMS a MMS zprávami. Nicméně při vytváření aplikace je nezbytné detailně si prostudovat dokumentaci a všechny specifikace.

Nastavení Android projektu pro práci s SMS a MMS zprávami je poměrně jednoduché a existuje několik návodů a postupů. Jednoduchý a srozumitelný návod je např. v repositáři android-smsmms na GitHubu⁴.

5.1 MMS PDU v Android API

Každý PDU balíček pro MMS zprávy ze specifikace 3GPP má v Javě (resp. v Android API) vlastní třídu. Tabulka 1 ukazuje příslušné kombinace. V první sloupci je název balíčku PDU dle specifikace 3GPP. Ve druhém sloupci je název příslušné třídy v Javě, který odpovídá PDU balíčku ze specifikace. Ve třetím sloupci jsou názvy konstant, reprezentující jednotlivé balíčky (více v kapitole 5.1.1). Ve čtvrtém sloupci je uveden směr přenosu balíčku z pohledu mobilního zařízení, tedy zdali je daný balíček odeslán nebo přijímán od MMS centra. Každá třída reprezentující PDU balíček má příslušné metody pro nastavování a čtení atributů jednotlivých PDU balíčků, které reprezentuje.

Tabulka 1 MM1 balíčky a třídy v Javě

Typ MM1 balíčku	Třída v Javě	Konstanta reprezentující balíček	Směr přenosu
MM1_submit.REQ	SendReq	MESSAGE_TYPE_SEND_REQ	Odeslat
MM1_submit.RES	SendConf	MESSAGE_TYPE_SEND_CONF	Přijmout
MM1_notification.REQ	NotificationInd	MESSAGE_TYPE_NOTIFICATION_IND	Přijmout
MM1_notification.RES	NotifyRespInd	MESSAGE_TYPE_NOTIFYRESP_IND	Odeslat
MM1_retrieve.RES	RetrieveConf	MESSAGE_TYPE_RETRIEVE_CONF	Přijmout
MM1_acknowledgement.REQ	AcknowledgeInd	MESSAGE_TYPE_ACKNOWLEDGE_IND	Odeslat
MM1_delivery_report.REQ	DeliveryInd	MESSAGE_TYPE_DELIVERY_IND	Přijmout
MM1_read_reply_recipient.REQ	ReadReclnd	MESSAGE_TYPE_READ_REC_IND	Odeslat
MM1_read_reply_originator.REQ	ReadOrigInd	MESSAGE_TYPE_READ_ORIG_IND	Přijmout

⁴ <https://github.com/klinker41/android-smsmms>

5.1.1 Pomocné třídy pro PDU balíčky

Vedle tříd pro PDU balíčky obsahují zdrojové kódy i několik dalších pomocných tříd. Tyto třídy jsou nezbytné pro snadnou práci s PDU balíčky a pro lepší pochopení problematiky jsou některé z nich popsány níže. Následující informace vycházejí ze zdrojových kódů. [20]

5.1.1.1 GenericPdu

Třída `GenericPdu` je базovou třídou pro všechny třídy reprezentující PDU balíčky. Tato třídy obsahuje základní informace o balíčku jako je typ, verze a odesílatel. Také má atribut typu `PduHeaders`, která obsahuje veškeré parametry PDU.

5.1.1.2 PduHeaders

Třída `PduHeaders` slouží pro uchování všech parametrů jednotlivých PDU balíčků. Obsahuje instanci `HashMap`, která drží klíč a hodnotu jednotlivých parametrů PDU. Tato třída také obsahuje konstanty použité pro příslušné klíče a hodnoty jednotlivých parametrů.

Například konstanta `MESSAGE_TYPE` představuje klíč pro typ PDU balíčku a k tomu lze přiřadit hodnoty jednotlivých typů (viz Tabulka 1 třetí sloupec). Dobrým příkladem je také priorita zprávy. Konstanta `PRIORITY` je zde klíčem parametru a k ní lze přiřadit hodnoty z konstant `PRIORITY_LOW`, `PRIORITY_NORMAL` nebo `PRIORITY_HIGH`.

Třída má i několik metod pro nastavování a čtení parametrů PDU. V metodách pro nastavení parametru se kontroluje, zdali je daná hodnota přípustná pro daný klíč, což slouží jako základní validace.

5.1.1.3 PduBody a PduPart

Za zmínku také stojí třída `PduBody`, která je pouze v třídách `SendReq` a `RetrieveConf` a představuje samotný obsah zprávy (text nebo multimediální soubory). Každá položka obsahu zprávy je instancí třídy `PduPart`. `PduPart` drží konkrétní informaci o jejím názvu, typu a také samotná data.

5.1.1.4 PduComposer

Třída `PduComposer` a její metoda `make` převádí instanci z třídy PDU balíčku na pole bytů (`byte[]`), které se později odesílá jako zpráva. Třída, resp. metoda umí převést

pouze balíčky, které se odesílají (viz Tabulka 1 čtvrtý sloupec). Při převodu se také kontroluje validita, tedy jestli jsou všechny povinné parametry vyplněny a jestli mají očekávané a přípustné hodnoty.

5.1.1.5 PduParser

Další třídou pro práci s PDU balíčky je PduParser. Ta má metodu parse, která převádí pole bytů (byte[]) na instanci třídy PduHeaders, která je následně využita pro vytvoření instance třídy konkrétního PDU balíčku. Metoda vrací instanci báze třídy GenericPdu, kterou je možné podle uvedeného typu přetypovat na konkrétní instanci třídy PDU balíčku.

5.1.1.6 SMIL a SmilHelper

SMIL (Synchronized Multimedia Integration Language) je jazyk založený na XML vytvořený konsorciem W3C. SMIL má za úkol popsat prezentaci multimediálního obsahu (textu, obrázků, videa, zvuku nebo odkazů). Definuje značky pro časování, rozložení, animace, vizuální přechody a vkládání médií. [21]

Dříve nebylo možné na mobilních zařízeních implementovat všechny funkce SMIL, proto byla vytvořena odnož nazvaná MMS SMIL. Tu bylo možné snadno implementovat na všech zařízeních. MMS SMIL dnes spravuje OMA. [2]

Třída SmilHelper⁵ je součástí balíčku com.google.android.mms.smil. Jediným účelem této třídy je převést vytvořenou instanci třídy PduBody ze všemi součástmi zprávy na objekt typu SMILDocument, který je potomkem třídy Document, tedy XML dokumentem. Tento objekt (XML dokument) se pak přidá do stejného PduBody, z jakého byl vytvořen, a to je následně vloženo do PDU balíčku SendReq.

⁵ <https://github.com/klinker41/android-smsmms/blob/master/library/src/main/java/com/google/android/mms/smil/SmilHelper.java>


```

<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
  <head>
    <layout />
  </head>
  <body>
    <par dur="8000ms">
      <text src="1521987829862.txt" />
      
    </par>
    <par dur="8000ms">
      <audio src="4968397717808649074.mp3" />
    </par>
  </body>
</smil>

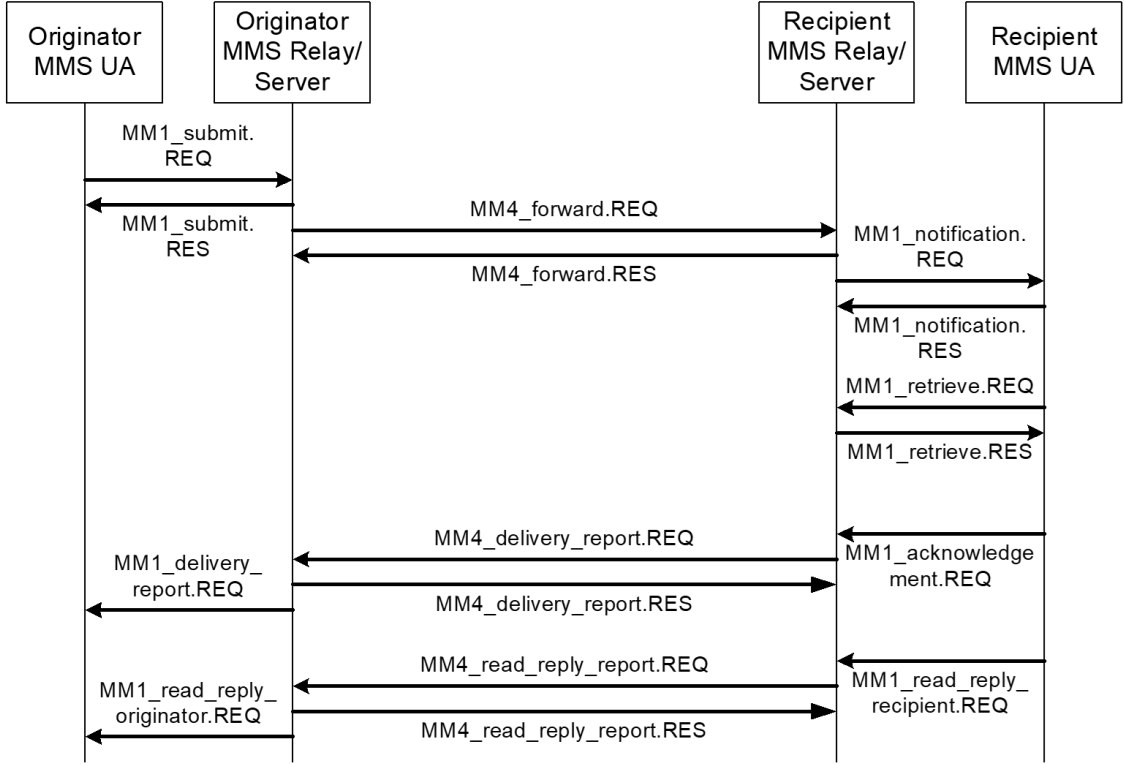
```

Ukázka kódu 1 Vygenerovaný SMIL

Ukázka kódu 1 ukazuje vygenerovaný SMIL pro PduBody, které má PduPart typu text, obrázek a zvuk. Atributy src obsahují cesty k souborům uložených v PduPart. Atribut dur označuje délku prezentace multimediálního objektu. Třída SmilHelper toto trvání nastavuje ve výchozím stavu na osm sekund, tedy 8000 milisekund. [22]

5.2 Scénář práce s PDU třídami

Pro pochopení sekvence odesílání a přijímání PDU balíčků zprávy, je velice důležitý Obrázek 5 ze specifikace 3GPP. Ten zobrazuje veškerou komunikaci zařízení s MMS centrem, aby došlo k odeslání a úspěšnému přijetí zprávy.



Obrázek 5 Sekvence odesílání a přijímání PDU balíčků [12]

5.2.1 Message ID

Parametr Message ID je jeden z mnoha parametrů PDU balíčků. Přiřazuje ho MMS centrum po přijetí nové zprávy (resp. balíčku SendReq). Parametr slouží koncovým zařízením i MMS centru pro identifikaci PDU balíčků, které patří té samé zprávě. Typickým příkladem je použití pro zprávy o přečtení nebo doručení. Tyto zprávy mohou být doručeny až po určitém čase a musí být jasně definované, které odeslané zprávě právě patří. Je důležité, aby MMS centrum i aplikace uchovávali Message ID přijatých PDU balíčků a správě jej nastavovali balíčkům odesílaným. Udržuje se tak jakýsi kontext v sekvenci balíčků. [12]

5.2.2 Odeslání zprávy

Odesílatel zprávy (Originator) vytvoří balíček typu MM1_submit.REQ (instance třídy SendReq). Instanci se nastaví veškeré povinné parametry a samotný obsah zprávy (text a multimediální obsah) v podobě PduBody. Tato instance se pomocí třídy PduComposer převede na pole bytů, které se odešle do MMS centra mobilního operátora.

Výsledkem tohoto přenosu je pole bytů (jako odpověď MMS centra), které obsahuje MM1_submit.RES. To lze převést na instanci typu SendConf pomocí třídy PduParser. Instance obsahuje informaci, zdali byla zpráva úspěšně přijata, a také identifikátor zprávy (Message ID), pomocí něhož později odesílatel identifikuje zprávu o přijetí nebo přečtení.

5.2.3 Přijetí zprávy

O další zpracování se stará MMS centrum. To pomocí rozhraní MM4 odešle zprávu do MMS centra operátora příjemce. Příjemci je pak doručen balíček MM1_notification.REQ, který jej má upozornit, že má k dispozici novou zprávu. Jedná se o instanci třídy NotificationInd, která obsahuje základní informace o zprávě jako je odesílatel, typ zprávy nebo datum a čas expirace. Nejdůležitějším parametrem je umístění zprávy, pomocí něhož si může příjemce stáhnout celou zprávu.

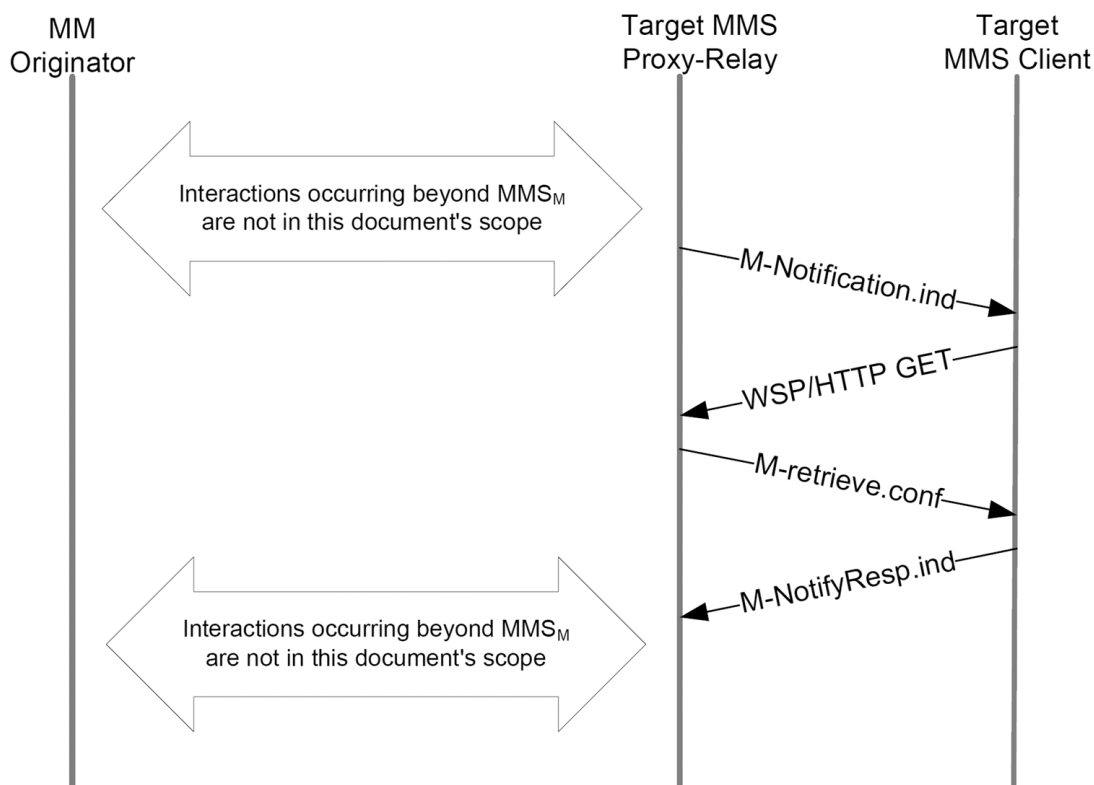
Celá zpráva je v PDU balíčku typu MM1_retrieve.RES reprezentovaná instancí třídy RetrieveConf. Tato instance obsahuje všechny informace o zprávě – odesílatele,

všechny příjemce, třídu a prioritu zprávy, datum a čas expirace atd.⁶ Nejdůležitější je ale parametr PduBody, ve kterém je uložen obsah zprávy. Příjemce z něho může získat text zprávy a všechny multimediální soubory poslané ve zprávě.

5.2.4 Okamžité nebo odložené stažení

Po obdržení MM1_notification.REQ si příjemce (resp. aplikace příjemce) může vybrat, zdali zprávu stáhne z MMS centra ihned nebo až na pokyn uživatele.

Jak ukazuje Obrázek 6, aplikace příjemce by měla po okamžitém stažení zprávy vytvořit a odeslat balíček MM1_notification.RES (instance třídy NotifyRespInd). Instanci je třeba nastavit Message ID přijaté zprávy a status RETRIEVED. To informuje MMS centrum, že zpráva byla úspěšně přijata a může odesílatelovi odeslat zprávu o přečtení (pokud je vyžadována a povolena).

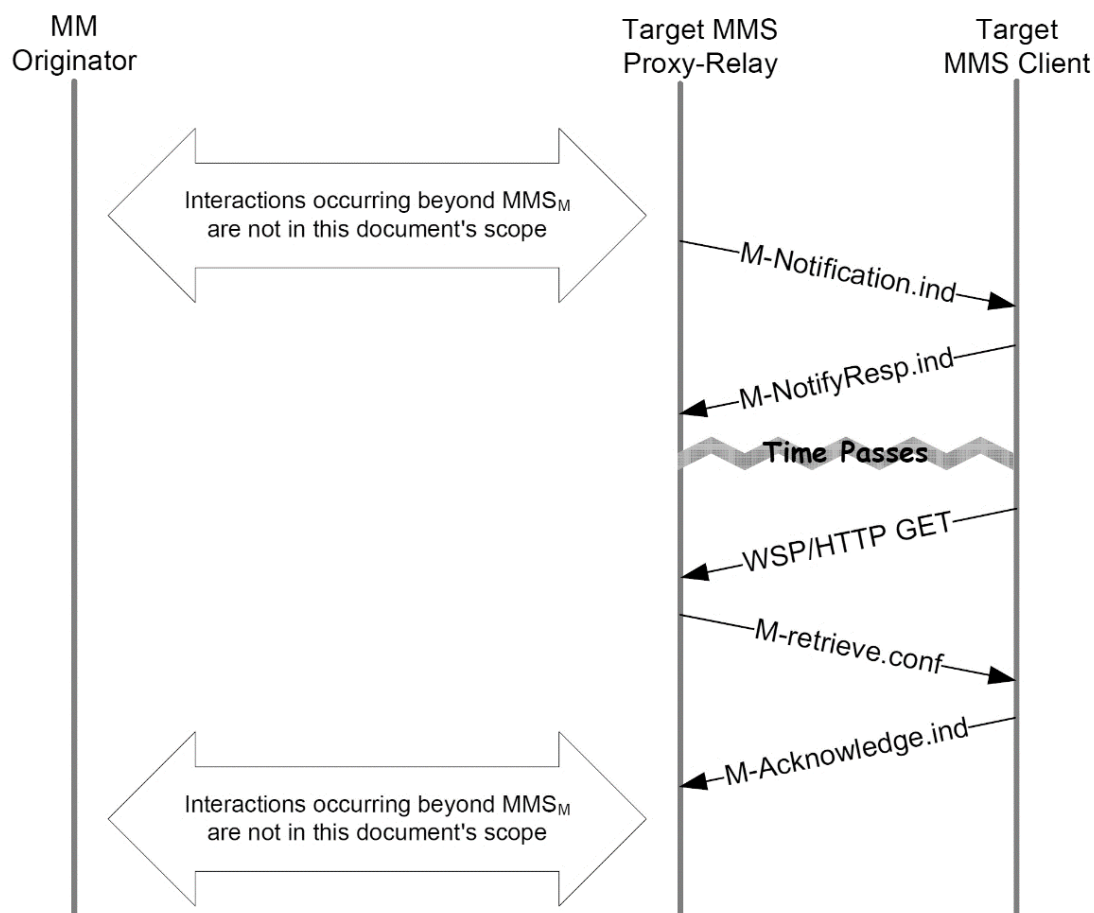


Obrázek 6 Okamžité stažení MMS zprávy [13]

Obrázek 7 zobrazuje scénář pro odložené stažení. Po přijetí oznámení o nové zprávě, příjemce odešle opět MM1_notification.RES s Message ID, ale nastaví status na DEFERRED. Po nějakém čase, až se příjemce rozhodne zprávu stáhnout, vytvoří se požadavek na stažení. A po úspěšném přijetí a zpracování příjemce odešle PDU typu

⁶ Kompletní přehled parametrů je ve specifikaci 3GPP [12]

MM1_acknowledgement.REQ opět s nastaveným Message ID zprávy. Tento scénář oznamuje MMS centru, že se jedná o odložené stažení a centrum tedy odešle zprávu o přečtení až po obdržení balíčku z třídy AcknowledgeInd.



Obrázek 7 Odložené stažení MMS zprávy [13]

5.2.5 Zprávy o doručení a přečtení

Požadavek na zprávu o doručení nebo zprávu o přečtení definuje odesílatel při nastavování parametrů instance SendReq před odesláním. Pokud to příjemce nezakáže (v parametru ReportAllowed balíčků NotifyRespInd nebo AcknowledgeInd) je úkolem MMS centra tyto zprávy doručit. Zpráva o přečtení je reprezentována balíčkem MM1_delivery_report.REQ v instanci třídy DeliveryInd. Tento balíček opět obsahuje Message ID, pomocí něhož může odesílatel zpětně identifikovat ke které zprávě zpráva o přečtení patří. Také je zde datum a stav doručení.

O přečtení zprávy musí informovat aplikace příjemce. Vytvoří PDU typu MM1_read_reply_recipient.REQ (třída ReadRecInd) a nastaví Message ID, datum

a čas přečtení. Také je zde status, který může být buď Přečteno nebo Smazáno bez přečtení. Tento PDU balíček odešle na své MMS centrum podobně jako běžnou zprávu. Původnímu odesílateli je pak doručen balíček typu MM1_read_reply_originator.REQ (ReadOrigInd), ve kterém jsou parametry tak, jak je nastavil příjemce (odesílatel zprávy o přečtení) a informuje tak odesílatele o přečtení nebo smazání zprávy a času, kdy se tak stalo.

5.3 Nastavení aplikace pro práci se zprávami

Mobilní zařízení se systémem Android může mít vždy pouze jednu aplikaci vybranou jako výchozí pro SMS a MMS zprávy. Tato funkce byla přidána od verze Android 4.4 a má zjednodušit práci vývojářům a umožnit uživatelům přehlednější nastavení. [23]

Pomocí statické metody `getDefaultSmsPackage` třídy `Telephony` lze jednoduše zjistit, zdali je aplikace právě označena jako výchozí. Funkce vrací textový řetězec obsahující název balíčku výchozí aplikace. Takto lze zjistit, zdali je naše aplikace právě označena jako výchozí pro SMS a MMS zprávy. Ukázka kódu 2 ukazuje, jak lze zobrazit systémový dialog pro přepnutí výchozí aplikace.

```
Intent intent = new Intent(context, Telephony.Sms.Intents.ACTION_CHANGE_DEFAULT);
intent.putExtra(Telephony.Sms.Intents.EXTRA_PACKAGE_NAME,
    context.getPackageName());
startActivity(intent);
```

Ukázka kódu 2 Zobrazení dialogu pro přepnutí výchozí aplikace

Pouze výchozí aplikace pro zprávy obdrží signál `SMS_DELIVER_ACTION`, když uživatel dostane novou SMS zprávu a `WAP_PUSH_DELIVER_ACTION`, když uživatel dostane novou MMS zprávu. [24] Každá aplikace pro zprávy musí umět tyto signály zpracovávat. Dokumentaci a postup pro zpracování signálů lze najít v oficiální dokumentaci, blogu pro Android vývojáře⁷ nebo i s ukázkovým příkladem na GitHubu⁸. Každá aplikace by měla umět zprávy nejen přijímat a zpracovávat, ale i vytvářet a odesílat. Pro odeslání zprávy musí být aplikace opět označena jako výchozí a musí dostat patřičná oprávnění (opět detailně popsáno v dokumentaci nebo v ukázkovém příkladu⁹).

⁷ <https://android-developers.googleblog.com/2013/10/getting-your-sms-apps-ready-for-kitkat.html>

⁸ <https://github.com/klinker41/android-smsmms>

⁹ <https://github.com/klinker41/android-smsmms>

5.4 Rozdíly v MMS API na různých verzích Android

Od verze Android 6.0 došlo k podstatné změně v API pro práci s MMS zprávami. Hlavní změny spočívají v práci s nastavením APN a metodami odesílání a přijímání zpráv.

5.4.1 APN a Subscription ID

Jak je zmíněno v kapitole 4.5, každá SIM karta obsahuje základní údaje o mobilním operátorovi, pomocí nichž se nastaví výchozí APN telefonu. Při odesílání nebo přijímání zprávy je nutné toto nastavení načíst a použít jej pro připojení k MMS centru.

Od verze Android 4.2 není možné číst APN nastavení, bez speciálního oprávnění „WRITE_APN_SETTINGS“. Toto oprávnění mohou mít pouze aplikace, které jsou v zařízení označeny jako systémové. Lze ale získat čísla MCC a MNC přímo ze SIM karty, pomocí nichž lze z databáze načíst předdefinované APN pro konkrétního mobilního operátora.

Od verze Android 6.0 se s APN nastavením pracuje pouze nepřímo. Pokud je aplikaci uděleno oprávnění „READ_PHONE_STATE“, může pomocí jednoduchého volání (Ukázka kódu 3), získat seznam všech SIM karet vložených do telefonu (mnoho telefonů dnes podporuje i více než jednu SIM kartu). [23]

```
SubscriptionManager subscriptionManager = SubscriptionManager.from(context);  
List<SubscriptionInfo> subscriptionInfos =  
    subscriptionManager.getActiveSubscriptionInfoList();
```

Ukázka kódu 3 Získání seznamu aktivních SIM

Jedná se o seznam instancí třídy SubscriptionInfo, které obsahují detailní informace o SIM kartě (ICCID, MCC, MNC, do jakého slotu je vložena). Klíčovým atributem je ale tzv. SubscriptionID. Tato hodnota jednoznačně identifikuje SIM kartu v telefonu a je použita při odesílání zprávy jako identifikátor, která SIM karta se má použít pro tuto službu. Aplikace mají často implementováno nastavení, kde si uživatel vybere, jakou SIM kartu (pokud jich má více) chce použít pro odeslání konkrétní zprávy. Při přijetí zprávy tento identifikátor určuje, na jakou SIM kartu (resp. od jakého operátora) byla zpráva přijata.

Subscription ID tedy velmi zjednodušuje vývoj komunikačních aplikací, protože se vývojář již nemusí starat o nějaké adresy MMS center, ale má pouze seznam aktivních SIM karet, ze kterých vybere jednu podle identifikátoru. [25]

5.4.2 API pro odeslání MMS

Tato kapitola navazuje na kapitolu 5.2.1, kde bylo popsáno vytváření obsahu zprávy a odeslání zmíněno jen abstraktně.

Samotné odeslání zprávy se v systému Android také liší podle verze.

Před verzí Android 6.0 (API level 23) se odesílání a příjem zpráv provádělo pomocí HTTP připojení. Pomocí APN adresy se vytvořila instance HTTPClient, která data odeslala jako POST požadavek na MMS centrum. Odeslání vytvořilo HTTPResponse, ve kterém byl výsledek požadavku a případně i odpověď MMS centra (například konkrétní data SendConf balíčku). [26]

Od verze Android 6.0 lze použít třídu SmsManager z balíčku android.telephony. Tato třída má potřebné metody pro odesílání SMS a MMS zpráv. Metodě se pouze předá obsah zprávy a volitelné dodatečné nastavení a o vše se postará systémová služba, který je spuštěna automaticky uvnitř těchto metod. [24]

5.4.3 API pro přijetí a stažení MMS

Signály o přijetí nové zprávy (popsané v kapitole 5.3) obdrží třída, která je potomkem abstraktní třídy BroadcastReceiver a je nastavena pro daný signál v manifestu aplikace¹⁰.

Je vhodné mít dvě oddělené třídy pro příjem SMS a MMS zpráv. Třída BroadcastReceiver má abstraktní metodu onReceive s parametry Context a Intent. Tato metoda je automaticky zavolána po přijetí nové zprávy od MMS centra. Ukázka kódu 4 ukazuje, jak získat potřebná data. V instanci třídy Intent předané v parametru metody jsou veškeré informace o přijaté zprávě. Zde je i identifikátor SubscriptionID, tedy na jakou SIM kartu byla zpráva přijata. Dále instance třídy Intent obsahuje data zprávy. Jedná se pole bytů obsahující PDU balíček zprávy. Po použití třídy PduParser a metody parse (popsané v 5.1.1), lze rozhodnout, jaký PDU balíček byl právě přijat. [26]

¹⁰ Více o nastavení: <https://github.com/klinker41/android-smsmms>

```

public class MmsListener extends BroadcastReceiver {

@Override
public void onReceive(Context context, Intent intent) {
    int subscriptionId = intent.getExtras().getInt("subscription", -1);
    byte[] data = intent.getByteArrayExtra("data");
    // spuštění asynchronního zpracování dat
}
}

```

Ukázka kódu 4 Metoda onReceive

```

GenericPdu pdu = PduByteHelper.getPdu(data);
if (isNotification(pdu)) {
    // Přijetí oznámení o nové zprávě
    NotificationInd pduNotificationInd = (NotificationInd)pdu;
}
else if (isDeliveryInd(pdu)) {
    // Přijetí zprávy o doručení
    DeliveryInd deliveryInd = (DeliveryInd)pdu;
}
else if (isReadOrigInd(pdu)) {
    // Přijetí zprávy o přečtení
    ReadOrigInd readOrigInd = (ReadOrigInd)pdu;
}
}

```

Ukázka kódu 5 Zpracování přijatého PDU balíčku

Ukázka kódu 5 pak ukazuje, jak zpracovat přijatý PDU balíček od MMS centra a určit jeho typ (NotificationInd, DeliveryInd a ReadOrigInd).

Po získání PDU balíčku typu NotificationInd je opět nutné rozlišovat mezi jednotlivými verzemi Android. Ve starších verzích Android před verzí 6.0 je opět nutné vytvořit HTTP připojení, které pomocí GET požadavku s adresou umístění (Content Location) stáhne MMS zprávu z MMS centra (respektive pole bytů s PDU balíčkem RetrieveConf).

Od verze Android 6.0 je to jednodušší, stačí vytvořit instanci třídy SmsManager a zavolat metodu downloadMultimediaMessage, která v parametru mimo jiné požaduje adresu dat na MMS centru (Content Location) a SubscriptionID. [24]

5.4.4 Práce s RR a DR

Odesílání PDU balíčku vytvořených z instancí tříd NotifyRespInd, AcknowledgeInd a ReadRecInd probíhá úplně stejně, jako odeslání samotné zprávy (odeslání balíčku SendReq). Vytvoří se instance dané třídy, převede se na pole bytů pomocí třídy PduComposer a odešle se stejným postupem popsáním v kapitole 5.4.2 v závislosti na verzi Android.

Jak již bylo zmíněno, PDU balíčky typu DeliveryInd a ReadOrigInd jsou přijímány stejně jako balíček NotificationInd. Rozdíl je v tom, že jsou kompletně přijaty již

v metodě `onReceive` ve třídě, která je potomkem třídy `BroadcastReceiver`. Není tedy nutné volat další funkce pro stažení jako je tomu u `NotificationInd`.

5.5 Možnosti ukládání zpráv

Pro ukládání SMS a MMS zpráv jsou v zásadě dvě možnosti. První možnost je použít systémovou databázi, do které mimo jiné ukládá zprávy předinstalovaná systémová aplikace. Druhou možností je implementovat vlastní ukládání (soubory, databáze atd.). [24]

Obě možnosti mají své výhody i nevýhody. Výhodou systémové databáze je, že i po odinstalování aplikace jsou uložené zprávy nadále k dispozici. Tato databáze slouží jako sdílené úložiště pro všechny aplikace, které ji chtějí využít. Na rozdíl od toho implementace vlastního řešení poskytuje jistou flexibilitu a další možnosti práce s uloženými záznamy. Díky tomu některé aplikace poskytují například možnost šifrování uložených zpráv, což může sloužit jako bezpečnostní prvek, při ztrátě telefonu. Nevýhodou je, že jsou tato data často uložena v adresáři pro data aplikace, který se automaticky maže po odinstalování aplikace. Také není žádná jiná možnost, jak se dostat k uloženým zprávám než použít právě aplikaci, která je vytvořila.

Obecně se pro práci se systémovou databází používá rozhraní `Content Provider`¹¹. Ukázka kódu 6 ukazuje volání, pro načtení všech odeslaných SMS zpráv seřazených podle data.

```
Cursor query = context.getContentResolver().query(
    Uri.parse("content://sms/sent"), null, null, null, "date desc");
```

Ukázka kódu 6 Načtení odeslaných SMS

Součástí Android API pro práci se zprávami je i třída `PduPersister`. Ta poskytuje metody pro práci se systémovou databází pro MMS zprávy a jejich části (`PduBody` a `PduPart`). [20] Ukázka kódu 7 ukazuje volání, pro uložení instance PDU balíčku `SendReq` do databáze odeslaných MMS zpráv.

```
PduPersister persister = PduPersister.getPduPersister(context);
Uri messageUri = persister.persist(
    sendReq, Uri.parse("content://mms/outbox"), true, true, null);
```

Ukázka kódu 7 Uložení odchozí MMS zprávy

¹¹ <https://developer.android.com/guide/topics/providers/content-providers.html>

6 Vývoj mobilní aplikace

Jedním z cílů této práce je vytvořit a popsat mobilní aplikaci pro operační systém Android, která bude sloužit pro podporu testování MMS center a bude umožňovat nastavení různých parametrů MMS zprávy a parametrů odesílání nebo přijímání. Tomuto vývoji se budou věnovat následující kapitoly.

6.1 Zadání a seznam požadavků

Požadavkem bylo vytvořit mobilní aplikaci pro operační systém Android. Aplikace má implementovat standardní funkcionalitu MMS klienta, a to komunikaci s MMS centrem (odesílání a příjem zpráv) rozšířenou o analýzu a možnost nastavení široké škály parametrů popsaných ve specifikaci 3GPP TS 23.140. Při implementaci bylo možné využít kromě standardního Android Google API například i open source projekt Signal-Android od WhisperSystems.

Dalším požadavkem bylo definování několika profilů odesílání zpráv a možnost uložit zprávy jako šablony a odeslat je později nebo opakovaně. U každé MMS zprávy se dají zobrazit její detaily – všechny PDU a parametry. Pokud odesílatel vyžadoval zprávu o doručení nebo zprávu o přečtení, a ta přišla, bude také možné zobrazit její detail. Lze odeslat zprávy s přílohou velikosti až 3 MB, zprávu s nastavenou dobou expirace nebo zprávu se skrytým odesílatelem.

6.2 Rešerše existujících řešení

Přestože existuje velké množství aplikací umožňujících komunikaci pomocí SMS a MMS zpráv, jen málo z nich má veřejně dostupné zdrojové kódy.

6.2.1 Knihovna android-smsmms

Knihovnu android-smsmms vytvořil Jake Klinker a umístil ji veřejně do systému GitHub. Tato knihovna tvoří zjednodušené API nad původním Android API pro práci se zprávami.

Použití knihovny je velice snadné a celý návod je popsán přímo v repositáři projektu. Stačí importovat knihovnu, přidat do projektu uvedená oprávnění a provést drobné úpravy v souboru manifestu aplikace. Dále lze v tomto repositáři nalézt ukázkový projekt nastiňující práci s knihovnou.

Knihovna obsahuje třídu Settings, která slouží pro nastavení několika parametrů zpráv. Možnosti nastavení parametrů zpráv jsou ale naprosto nedostatečné vzhledem k zadání aplikace. Ukázka kódu 8 nastiňuje, jak vytvořit nastavení, zprávu a vše odeslat.

```
Settings settings = new Settings();  
// nastavení základních parametrů zprávy  
  
Transaction transaction = new Transaction(mContext, settings);  
Message message = new Message(messageText, addressToSendTo);  
message.setImage(mBitmap);  
transaction.sendNewMessage(message, threadId)
```

Ukázka kódu 8 Nastavení parametrů zprávy v knihovně android-smsmms

Při detailním prozkoumání zdrojových kódů knihovny, lze najít část, kde se vytváří PDU balíček SendReq. Při vytváření je většina hodnot nastavena na nějakou výchozí hodnotu. Zprávy o přečtení zde nejsou řešeny vůbec.

Dostupná knihovna navíc tvoří pouze jakousi základní šablonu, veškeré uživatelské prostředí si pak každý musí vytvořit sám. Knihovna je použita v aplikacích od stejného autora EvolveSMS, Pulse nebo Sliding Messaging Pro.

I přes všechny nedostatky může tato knihovna sloužit jako dobrá inspirace pro vlastní řešení. Například ve třídě PushReceiver je velice dobře vidět větvení a zpracování příchozích PDU balíčků od MMS centra.

Knihovna také obsahuje zdrojové kódy tříd z oficiálního Android API pro zprávy. To může pomoci lepšímu pochopením vnitřního fungování logiky odesílání a přijímání zpráv. [24]

6.2.2 WhisperSystems/Signal-Android

Projekt Signal-Android představuje hotové řešení pro komunikaci pomocí SMS a MMS zpráv. Aplikace existuje pro systémy Windows, iOS i Android. V repositáři na GitHubu jsou kompletní zdrojové kódy aplikace.

Aplikace umožňuje kromě standartního odesílání a přijímání SMS a MMS zpráv také posílat zašifrované zprávy. Po registraci do systému Signal lze pozvat přátele a komunikovat s nimi v této aplikaci i pomocí běžného internetového připojení.

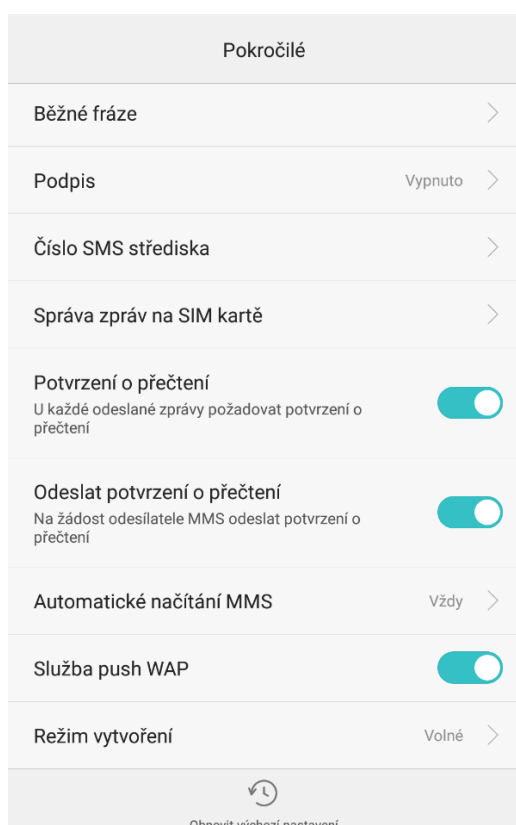
Aplikace Signal používá vlastní databázi pro ukládání zpráv (to je ale i nevýhoda, protože zprávy nejsou v systémové databázi a po odinstalování aplikace se mažou). Zprávy jsou uloženy šifrovaně, a dešifrují se pouze při zobrazení nebo při exportu.

Aplikace umožňuje importovat SMS zprávy ze systémové aplikace pro zprávy a také exportovat zprávy do XML souboru. Nicméně zde stále chybí několik prvků uvedených v zadání a nastavení vlastností zpráv je pouze omezené. [26]

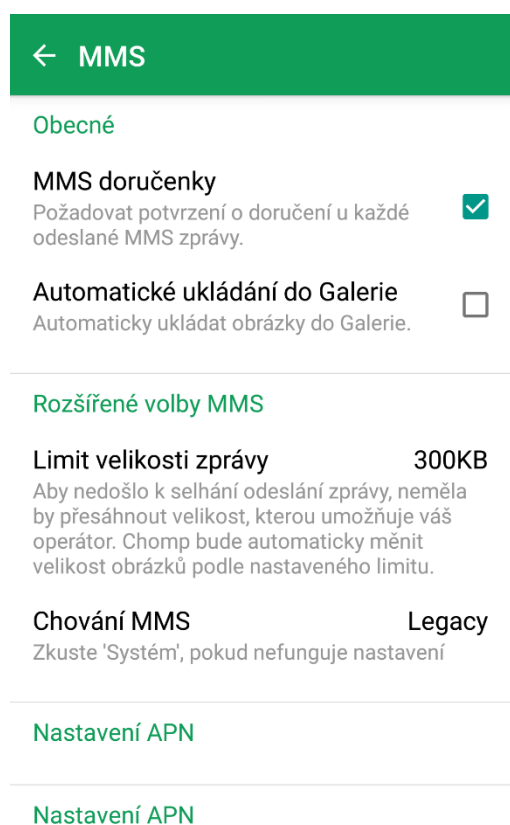
6.2.3 Výchozí aplikace na Android a podobné aplikace na Google Play

Každý mobilní telefon se systémem Android má již jednu aplikaci pro SMS a MMS zprávy předinstalovanou. Tato aplikace ve většině případů poslouží všem uživatelům velmi dobře. Aplikace se může lišit dle výrobce telefonu a verze systému Android.

Protože tyto výchozí předinstalované aplikace splňují všechny základní uživatelské požadavky pro běžného uživatele, lze je použít jako inspiraci pro vývoj vlastních aplikací. Avšak pro testování MMS center nejsou příliš vhodné, protože, jak ukazuje Obrázek 8, jejich nastavení, byť „Pokročilé“, je stále velice omezené. Lze zde nastavit požadování a odesílání zpráv o doručení a přečtení nebo automatické stažení příchozí MMS zprávy (kde je i možnost nestahovat v případě roamingu).



Obrázek 8 Nastavení ve výchozí aplikaci



Obrázek 9 Nastavení MMS v aplikaci Chomp SMS

V obchodě Google Play lze vyhledat a nainstalovat stovky podobných aplikací, sloužících pro komunikaci. Některé aplikace, podobně jak již zmíněná Signal, umožňují komunikovat pomocí internetového připojení, nebo zprávy různě šifrovat. Obrázek 9 ukazuje okno pro nastavení MMS zpráv v aplikaci Chomp SMS. Tato aplikace umožňuje oproti té výchozí nastavit i maximální velikost MMS zprávy nebo způsob odesílání. Některé aplikace dokonce umožňují nastavit odložené odeslání zprávy. Tato funkce je ale většinou implementována tak, že se odeslání odloží pouze z telefonu, aby jej mohl uživatel zrušit a nevyužije se tak atribut DeliveryTime v PDU balíčku SendReq.

7 Analýza a návrh aplikace

V rešerši existujících řešení je popsána jedna knihovna pro práci s SMS a MMS zprávami, jedna kompletní hotová aplikace s volně dostupnými zdrojovými kódy a výchozí aplikace bez zdrojových kódů. Nicméně ani jedna z variant nesplňuje všechny požadavky na aplikaci uvedené v zadání.

V obchodě Google Play ani na GitHubu není žádná aplikaci, která by umožňovala volně nastavovat všechny parametry MMS zpráv a pak je všechny přehledně zobrazit. Pokud aplikace umožňuje parametry zprávy zobrazit, jsou často velice omezené a nepředstavují kompletní přehled potřebný pro testování MMS center; je to často jen nezbytné minimum pro běžného uživatele.

7.1 Aplikace MMS Params

I přes některé nedostatky představuje projekt Signal-Android vhodné řešení, které bylo použito jako základ pro vlastní mobilní aplikaci nazvanou MMS Params. Projekt Signal-Android je pod licencí GPLv3, což umožňuje volné stažení, úpravy a následnou distribuci [27]. Vytvořená aplikace je opět pod licencí GPLv3 a je možné zobrazit a stáhnout její zdrojové kódy ze systému BitBucket¹². Aplikace je také zdarma ke stažení z obchodu Google Play¹³.

7.2 Analýza požadavků

Základní aplikace Signal-Android má kompletně implementované odesílání a příjem SMS a MMS zpráv. Aplikaci je třeba rozšířit o možnost nastavování parametrů MMS zpráv uživatelem. To bude řešeno seznamem profilů. V aplikaci lze definovat několik profilů, které budou obsahovat nastavení MMS zpráv, které se použije při jejich odesílání a zpracování po přijetí. Bude zde i výchozí profil, který bude použit v případě, kdy nebude žádný jiný profil vytvořen nebo vybrán jako aktivní.

Další funkcí aplikace bude seznam MMS šablon. Šablona bude obsahovat příjemce, vč. příjemců kopie a skryté kopie. Také zde bude možnost vložit předmět zprávy a samotný text včetně multimediálního obsahu (obrázky a soubory).

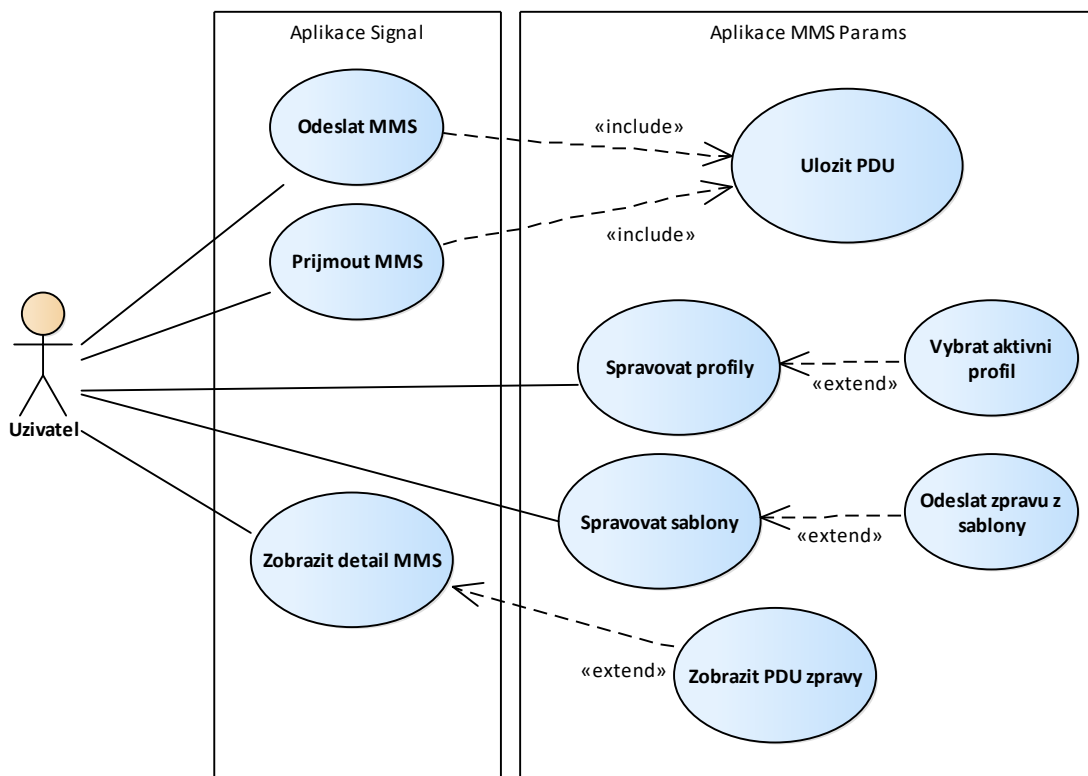
¹² <https://bitbucket.org/domestos99/mmsparams>

¹³ <https://play.google.com/store/apps/details?id=cz.uhk.bak.mmsparams>

Ke každé zprávě, odeslané i přijaté, se budou ukládat PDU balíčky, které byly při jejím zpracování buď vytvořeny a odeslány nebo přijaty z MMS centra. Aplikaci Signal je potřeba také rozšířit o možnost vložit ke zprávě více příloh, nyní aplikace umožňuje vložit nejvýše jednu.

7.2.1 Diagram případů užití

Obrázek 10 obsahuje navržený diagram případů užití. Případy užití aplikace Signal v diagramu jsou omezeny pouze na MMS zprávy. Je zde patrné, že aplikace umí MMS zprávy pouze odesílat, přijít a zobrazovat. V pravé části jsou nové případy užití, které budou implementovány v aplikaci MMS Params.



Obrázek 10 Diagram případů užití MMS Params

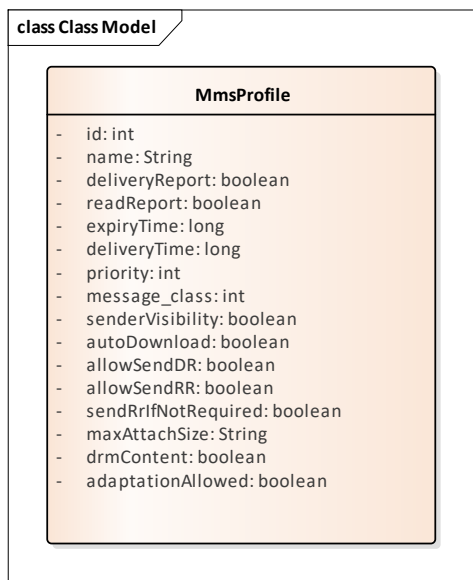
7.3 Profily nastavení

Profil obsahuje všechny parametry, které je možné nastavit odesílané zprávě. Vybraný profil také slouží jako nastavení chování aplikace. Toto nastavení zahrnuje, zdali přijatou zprávu stáhnout okamžitě nebo použít odložené stažení a stáhnout ji až na pokyn uživatele.

Je zde i nastavení, zdali povolit doručení zprávy o přečtení – to se použije při vytváření PDU NotifyRespInd a AcknowledgeInd. A nastavení zprávy o přečtení – uživatel může povolit nebo zakázat odeslání této zprávy na základě požadavku odesílatele.

Uživatel si v aplikaci může vytvořit několik profilů a vybrat jeden profil jako aktivní. Existuje zde i jeden profil označený jako výchozí, který se automaticky používá, pokud není žádný jiný profil vytvořen nebo vybrán jako aktivní.

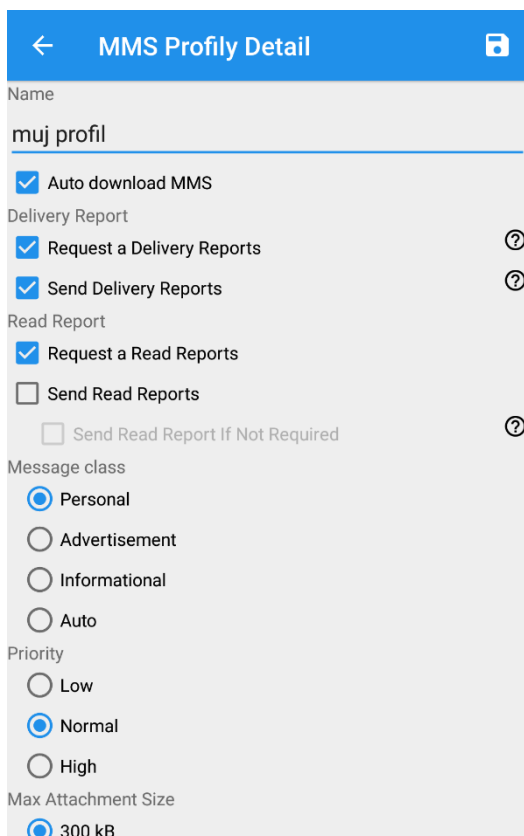
Obrázek 11 obsahuje UML třídu pro profil nastavení. Pro lepší práci s nastavením jsou některé atributy typu boolean. Tyto atributy se později při sestavování PDU balíčku převádějí na danou konstantu (např. hodnota True => PduHeaders.VALUE_YES), což je nezbytné, aby byl PDU balíček validní vzhledem ke specifikaci.



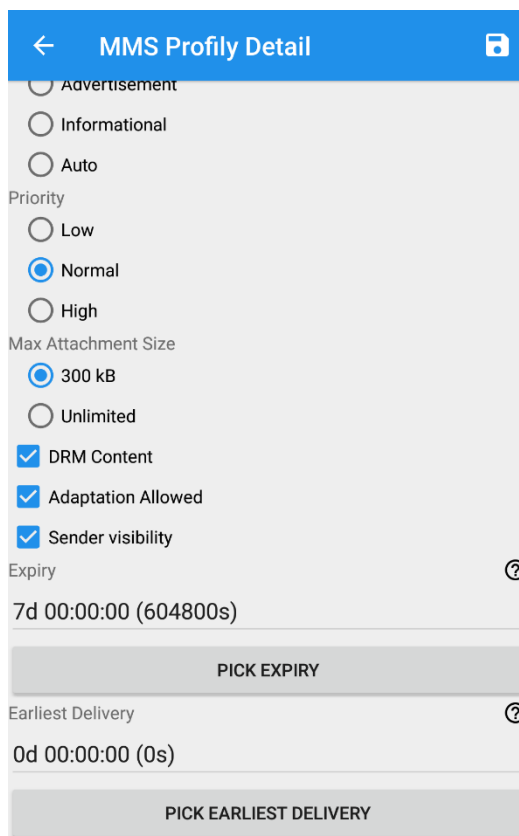
Obrázek 11 Třída MmsProfile

Každý vytvořený profil je uložen v databázi aplikace. Výchozí profil je uložen v preferencích aplikace společně s odkazem na právě vybraný aktivní profil.

Obrázek 12 a Obrázek 13 ukazují celou obrazovku editace detailu profilu. Uživatel si může každý profil pojmenovat a nastavit mu výše zmíněné parametry. U ukládání nebo načítání profilu jsou konstanty z atributů pro prioritu nebo třídu zprávy zpracovány a je označen patřičný přepínač hodnoty.

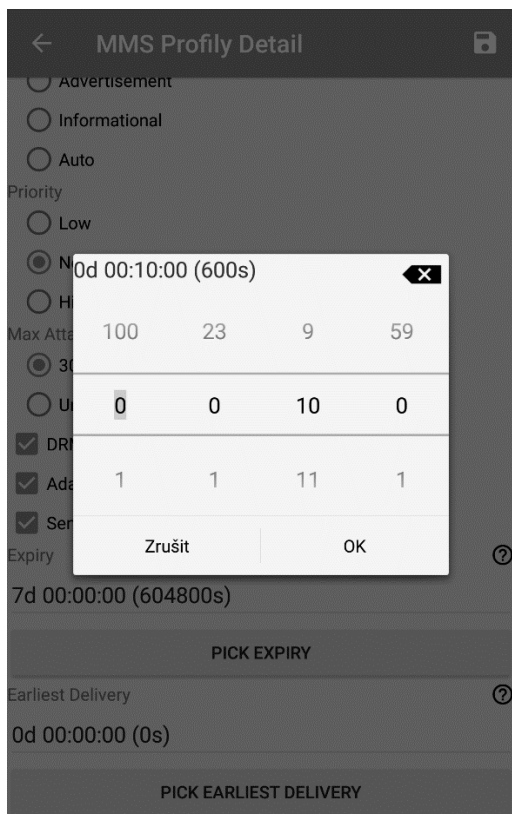


Obrázek 12 Detail nastavení profilu



Obrázek 13 Detail nastavení profilu

Obrázek 14 pak ukazuje modální okno pro výběr hodnoty času nejdřívejšího doručení zprávy. Stejný dialog je zobrazen i při výběru času expirace. Jedná se o relativní čas, který se začne počítat od chvíle odeslání zprávy. Vybraný čas je uložen v celých sekundách, protože právě v této jednotce je později nastavován do PDU balíčků.



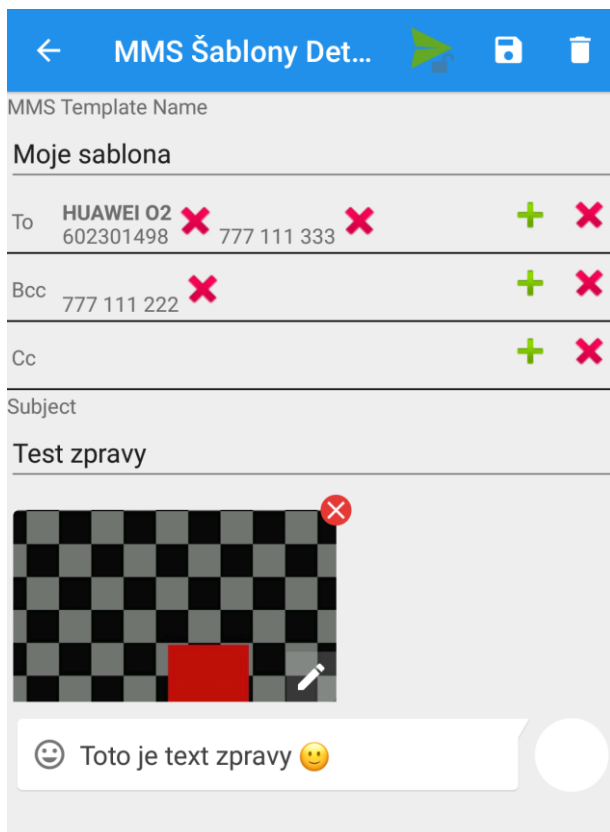
Obrázek 14 Výběr času expirace zprávy

7.4 MMS šablony

V aplikaci je seznam MMS šablon, kde si uživatel může předdefinovat zprávu, kterou pak bude odesílat opakovaně. MMS šablony mají na rozdíl od klasických zpráv další rozšířené možnosti. Lze vybrat více příjemců ze seznamu kontaktů nebo zadat číslo příjemce přímo. Je možné také zadat příjemce pro kopii zprávy nebo pro skrytou kopii. V neposlední řadě je zde políčko pro zadání předmětu MMS zprávy.

Nechybí ani editor samotného obsahu zprávy, tedy text zprávy a možnost výběru několika multimediálních souborů.

Po vytvoření šablony a kliknutí na tlačítko odeslat se šablona převede na normální MMS zprávu, nastaví se jí všechny potřebné parametry z aktivního profilu a zpráva se odešle. Odeslaná zpráva z šablony se zobrazí v klasické konverzaci jako odeslaná. Šablony tedy poskytují možnost efektivnějšího testování MMS center, protože je lze odesílat opakovaně a není tak nutné vytvářet stále novou zprávu ručně.



Obrázek 15 Detail nastavení MMS šablony

7.5 Zobrazení PDU balíčků zpráv

Při testování MMS center je nezbytné porovnávat parametry MMS zpráv, které mobilní zařízení odeslalo a které MMS centrum přijalo. Jak již bylo zmíněno, mobilní aplikace komunikuje s MMS centrem pomocí PDU balíčků. Ke každé zprávě je vytvářeno a přijímáno hned několik typů těchto balíčků.

Každý vytvořený (resp. odeslaný) a přijatý PDU balíček je ukládán do databáze s odkazem na konkrétní zprávu, ke které patří. V aplikaci je následně možné, zobrazit všechny PDU balíčky a jejich parametry, které patří k dané MMS zprávě.

Seznam balíčků je seřazen podle data a času jejich odeslání nebo přijetí. Každý balíček v seznamu balíčků zprávy má zobrazen jeho typ (viz kapitola 5.1) a každá položka obsahuje detailní seznam všech parametrů konkrétního PDU balíčku dle specifikace. V Android API jsou některé parametry PDU balíčku reprezentovány jako číselná konstanta. Tyto parametry jsou před zobrazením převedeny do textové podoby, aby uživatel jasně viděl, co daná hodnota reprezentuje.

Balíčky SendReq a RetrieveConf obsahují parametr PduBody, který se skládá ze seznamu částí typu PduPart (viz 5.1.1). V seznamu PDU balíčku je tedy u položek tohoto typu tlačítko, které zobrazí seznam částí v PduBody. U každé části je uveden její název, umístění, velikost a co obsahuje za data.

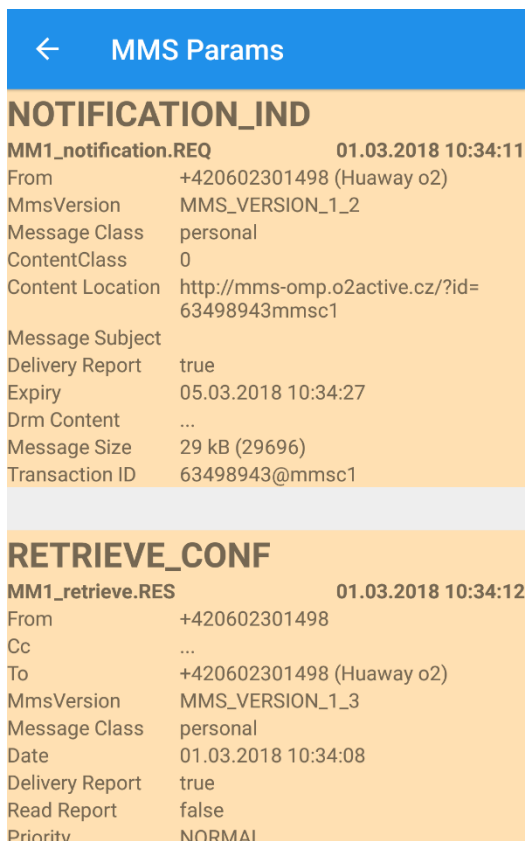
Obrázek 16 ukazuje, jak aplikace zobrazuje detail o PDU balíčku typu SendReq a Obrázek 17 ukazuje detail PduBody pro daný SendReq. Dále Obrázek 18 zobrazuje PDU balíček s informací o přijaté zprávě.



Obrázek 16 Detail PDU SendReq



Obrázek 17 Detail PduBody



Obrázek 18 Detail PDU pro přijatou zprávu

7.5.1 Výpočet velikosti zprávy

Balíčky SendReq a RetrieveConf neobsahují parametr pro velikost zprávy¹⁴. Velikost je tedy nutné vypočítat, aby uživatel věděl, kolik paměti každá zpráva zabírá. Výpočet je popsáný ve specifikaci a jedná se o součet velikosti parametru předmět zprávy a součet velikostí všech částí zprávy v PduBody kromě PduPart typu SMIL. Ukázka kódu 9 obsahuje metodu, pro výpočet velikosti jednoho PduPart (komentáře jsou uvedeny jako příklad hodnot, pokud se jedná o obrázek).

```
public static long getPartSize(PduPart part) {
    return part.getName().length // 123456789.jpg
        + part.getContentLocation().length // 123456789.jpg
        + part.getContentType().length // image/jpeg
        + part.getData().length // pole bytů obsahu
        + part.getContentId().length; // <123456789.jpg>
}
```

Ukázka kódu 9 Metoda pro výpočet velikosti zprávy

¹⁴ SendReq ho obsahuje, ale je třeba ho vypočítat.

7.6 Job Manager

JobManager je knihovna, jejíž zdrojové kódy jsou volně dostupné ke stažení v systému GitHub¹⁵. Knihovna umožňuje naplánovat úlohy, které jsou spuštěny buď okamžitě nebo až po splnění určitých podmínek.

Aplikace běžící na systému Android často potřebují provést nějaké operace, které mohou trvat déle nebo blokují určité zdroje. Tyto operace by měli běžet asynchronně, mimo hlavní pracovní vlákno. Může se jednat o různá volání webových služeb, kde odpověď serveru nemusí být vždy okamžitá nebo to může být stahování větších souborů nebo dat obecně. Běžně se používají služby (Services), asynchronní úlohy (AsyncTasks) nebo různá vlákna.

Standartní způsob v Android může zahrnovat spoustu časovačů, logiky pro opakování při neúspěchu, nemluvě o řešení závislostí. Knihovna JobManager řeší všechny tyto problémy. Lze vytvořit úlohy, které jsou jakousi jednotkou práce. Při jejich spuštění je možné nastavit řadu parametru, jako je podmínka spuštění, počet opakování při neúspěchu nebo povolené zdroje.

Každá úloha je serializována a uložena do uložiště aplikace, což zajišťuje, že je úloha spuštěna i když se aplikace restartuje před jejím spuštěním. Knihovna má také implementovaný jednoduchý systém pro Dependency Injection. Jednoduchým příkladem je Context. Instanci této třídy bude potřebovat většina spuštěných úloh, ale Context nelze uložit. V dokumentaci knihovny je jednoduchý návod, jak Dependency Injector nastavit. Použití této knihovny je velice jednoduché a je přehledně popsáno na GitHubu včetně kvalitně okomentovaných zdrojových kódů.
[28]

7.7 Firebase

Firebase je cloudová platforma vlastněná společností Google, která slouží pro usnadnění vývoje mobilní a webových aplikací. Platforma poskytuje mnoho služeb jako jsou databáze, autentizace uživatelů nebo hostování webových stránek. Hlavní myšlenkou je usnadnit vývoj aplikací na tolik, aby se vývojář nemusel vůbec starat o server nebo rozhraní API, ale mohl se plně zaměřit na uživatelský zážitek aplikace.

¹⁵ <https://github.com/signalapp/jobmanager>

Základní verze Firebase je zdarma a implementace do jakékoli aplikace je velmi jednoduchá, díky detailní a podrobné dokumentaci. [29]

7.7.1 Logování chyb na Firebase

V aplikaci MMS Params je Firebase použit hlavně pro logování chyb a pádů aplikace. Použitá služba se nazývá Crashlytics. Jedná se o odlehčený nástroj pro reportování chyb v reálném čase. Každé chybě lze přiřadit nějakou prioritu a vlastní kód (označení). Služba pak zobrazuje chyby přehledně ve skupinách spolu s informacemi o zařízeních, na kterých dané chyby vznikly a vykonaných akcích, provedených těsně před chybou. Lze také nastavit, že o každé nové chybě přijde ihned informace na email. [29]

7.7.2 Testy aplikace na Firebase

Platforma Firebase umožňuje provádět testy aplikace před jejím vydáním. Nástroj se nazývá Test Lab a využívá virtuální mobilní zařízení. Stačí pouze nahrát instalační soubor aplikace (v případě Androidu soubor apk) a vybrat zařízení a verze operačního systému, na kterých se má test provést.

Výsledkem testu je seznam chyb (pokud byly nějaké nalezeny), video z celého testu a také detailní graf zobrazující vytížení procesoru, paměti a síťového provozu během celého testu. [29]

7.8 Rozšiřitelnost aplikace

Nastavení zpráv a aplikace

Jak je popsáno v kapitole 7.3, třída MmsProfile obsahuje všechny parametry pro nastavení odesílané zprávy a chování aplikace. Pokud by byl v budoucnu požadavek na rozšíření aplikace o nastavování dalších parametrů zpráv nebo chování, přidali by se příslušné atributy do této třídy. Nastavení je uživateli prezentováno třídou MmsProfilesDetailFragment, do které by se přidaly další komponenty pro jednotlivé možnosti nastavení.

Šablony zpráv

Šablony zpráv jsou reprezentovány třídou `MmsTemplate`, která mimo jiné obsahuje nastavení příjemců, předmětu a obsahu zprávy. Prezenční vrstvu zajišťuje třída `MmsTemplateDetailFragment`. Bylo by možné zde přidat další atributy, které by dále zjednodušovali testování MMS center. Může se jednat například o nastavení automatického opakovaného odeslání a podobně.

Zobrazení detailu zpráv

Pro rozšíření zobrazovaných informací o zprávě lze použít třídu `MessageDetailsActivity`, která zobrazuje základní informace o zprávě jako je odesílatel nebo příjemce, datum odeslání nebo přijetí a velikost zprávy. Pro zobrazení jednotlivých PDU balíčků slouží třídy pojmenované `MessagePdu[název třídy balíčku]Fragment`. Každá z těchto tříd zobrazuje právě ty informace, které daný typ PDU balíčku obsahuje. Je tím zajištěna vyšší přehlednost a možnost nastavit zobrazované informace jednotlivě. Pokud by vznikl požadavek na zobrazení dalších detailů o balíčku, realizovalo by se to právě v těchto třídách. Detaily PDU balíčků `SendReq` a `RetrieveConf` umožňují zobrazit také jejich `PduBody`. To má na starost třída `MessagePduPduBodyListFragment`, která zobrazí seznam jednotlivých `PduPart`, kde každý z nich je zobrazován společnou třídou `MessagePduPduPartFragment`.

7.9 Problémy

Během analýzy, vývoje a testování aplikace se objevili některé problémy a omezení, které jsou popsány níže.

7.9.1 Chybějící parametry v PDU třídách

Ačkoli je Android API pro práci s MMS zprávami navrženo velmi dobře, některé parametry uvedené ve specifikaci 3GPP nejsou implementovány. Jedná se například o nastavení času nejdřívějšího doručení, nastavení `DrmContent` v třídě `SendReq` nebo parametr `ResponseText` ve třídě `SendConf`. Ve skutečnosti metody pro práci s těmito parametry v třídách existují, ale jsou zakomentovány a označeny jako nepodporované. Všechny další nepodporované parametry lze najít ve zdrojových

kódech tříd pro PDU balíčky. Tento problém byl vyřešen vytvořením potomka z dané třídy pro PDU balíček a implementací potřebných funkcí. [22]

7.9.2 Velikost PDU balíčků

Všechny PDU balíčky ke každé zprávě jsou ukládány do databáze aplikace. Databáze je poměrně robustní, co se týče množství uložených dat. Avšak pro splnění zadání byla implementována možnost, odesílat zprávy s přílohou takřka neomezené velikost. Při vytvoření SendReq nebo přijetí RetrieveConf jsou všechny multimediální soubory zprávy uloženy právě v tomto PDU balíčku, respektive v poli bytů uvnitř tohoto balíčku.

Pokud tedy uživatel odešle nebo přijme několik takovýchto zpráv s velkou přílohou, do databáze jsou ukládány poměrně velké záznamy. To dramaticky zvyšuje velikost databáze, a tedy i velikost dat uložených ve speciálním adresáři pro aplikaci. Na některých starších telefonech, obzvláště s menší interní pamětí, se může stát, že telefonu dojde paměť poměrně rychle. Řešením tohoto problému je pravidelně mazat starší zprávy, přičemž se s nimi mažou i všechny přiřazené záznamy PDU balíčků.

7.9.3 Nastavení APN v aplikaci

Jak již bylo popsáno v kapitole 5.4.1, od určité verze operačního systému Android již nelze volně nastavovat APN telefonu přímo z aplikací. Požadavek na toto nastavení tedy není splněn a uživatel musí APN nastavovat přímo v nastavení telefonu. Při testování MMS center jde hlavně o nastavování jiných adres než výchozích (produkčních). Zde se projevuje výhoda SubscriptionID v API, které ukazuje pouze na konkrétní SIM kartu a nastavené APN si telefon načte sám při odesílání nebo přijímání zprávy.

7.10 Jak aplikaci získat

Zdrojové kódy aplikace MMS Params jsou ve veřejném repositáři ve verzovacím systému BitBucket: <https://bitbucket.org/domestos99/mmsparams>. Aplikaci je možné najít a stáhnout přímo z obchodu Google Play z toho odkazu: <https://play.google.com/store/apps/details?id=cz.uhk.bak.mmsparams>

8 Závěr

Hlavním cílem této práce byl popis API pro práci s MMS zprávami na platformě Android a vytvoření mobilní aplikace pro podporu testování MMS center.

Popis obsahuje stručnou historii SMS a MMS zpráv, popis standardizace architektury mobilní sítě a rozhraní pro komunikaci. Vývojářská dokumentace pro MMS zprávy na platformě Android není příliš kvalitní, proto je nutné spoléhat se spíše na již existující a fungující řešení nebo prosté porozumění zdrojovým kódům. Aplikace byla implementována na platformě Android, kde se podařilo splnit stanovené cíle. Výsledná aplikace splňuje naprostou většinu požadavků specifikovaných v prvotní analýze. V průběhu vývoje aplikace a jejím následném testování se ukázaly ještě další požadavky ze strany zadavatele, které byly dodatečně doplněny. Jednalo se hlavně o rozšíření aplikace o některé parametry, které jsou sice ve specifikaci 3GPP, ale nebyly implementovány v Android třídách pro PDU balíčky.

Aplikace by měla být přínosem pro techniky mobilních operátorů, kteří mají na starost nastavování, správu a testování MMS center. Pomůže odhalit chybná nastavení center a zajistit tak kvalitní uživatelský zážitek při používání jejich služeb. Aplikace by se do budoucna mohla rozšířit o funkci plně automatizovaného testování včetně spouštění hromadných testů přímo z počítače a následného reportování výsledků a možných rizik nebo nesrovnalostí.

Seznam použité literatury

- [1] STATISTA. *Number of SMS and MMS sent in Germany from 1999 to 2014* [online]. [vid. 2018-03-25]. Dostupné z: <https://www.statista.com/statistics/461700/number-of-sms-and-mms-sent-per-day-germany/>
- [2] LE BODIC, Gwenaël. *Mobile messaging technologies and services: SMS, EMS and MMS*. B.m.: John Wiley & Sons, 2005. ISBN 0470014512.
- [3] 3GPP. *About 3GPP* [online]. [vid. 2018-03-15]. Dostupné z: <http://www.3gpp.org/about-3gpp>
- [4] 3GPP2. *About 3GPP2* [online]. [vid. 2018-03-15]. Dostupné z: https://www.3gpp2.org/Public_html/Misc/AboutHome.cfm
- [5] W3C. *W3C* [online]. [vid. 2018-03-15]. Dostupné z: <https://www.w3.org>
- [6] ADAPTIC. *Co je W3C. Adaptic* [online]. [vid. 2018-03-22]. Dostupné z: <http://www.adaptic.cz/znalosti/slovnicek/w3c/>
- [7] W3C. *Tim Berners-Lee* [online]. [vid. 2018-03-15]. Dostupné z: <https://www.w3.org/People/Berners-Lee/>
- [8] 3GPP. *Open Mobile Alliance and 3GPP* [online]. [vid. 2018-03-15]. Dostupné z: ftp://www.3gpp.org/PCG/PCG_09/DOCS/PDF/PCG9_22.pdf
- [9] OPEN MOBILE ALLIANCE. *OMA* [online]. [vid. 2018-03-15]. Dostupné z: <http://www.openmobilealliance.org/>
- [10] OMA. *About OMA* [online]. [vid. 2018-03-15]. Dostupné z: <http://openmobilealliance.org/about-oma>
- [11] OMA. *Frequently Asked Questions* [online]. [vid. 2018-03-15]. Dostupné z: <http://openmobilealliance.org/about-oma/frequently-asked-questions>
- [12] 3GPP. *Multimedia Messaging Service (MMS) TS 23.140* [online]. 2009. Dostupné z: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=794>
- [13] OMA. *Multimedia Messaging Service Client Transactions* [online]. 2005 [vid. 2018-03-15]. *OMA-TS-MMS-CTR-V1_3-20050927-C*. Dostupné z: http://www.openmobilealliance.org/release/MMS/V1_3-20050927-

C/OMA-TS-MMS-CTR-V1_3-20050927-C.pdf

- [14] WEIK, Martin H. protocol data unit. In: *Computer Science and Communications Dictionary* [online]. Boston, MA: Springer US, 2001, s. 1361. ISBN 978-1-4020-0613-5. Dostupné z: doi:10.1007/1-4020-0613-6_14994
- [15] BAIR, John. Chapter 6 - Subscriber Identity Module BT - Seeking the Truth from Mobile Evidence. In: [online]. B.m.: Academic Press, 2018, s. 71–85. ISBN 978-0-12-811056-0. Dostupné z: doi:https://doi.org/10.1016/B978-0-12-811056-0.00006-6
- [16] ŠIMKO, Andrej. *Forezní analýza SIM karet* [online]. Brno, 2012. b.n. Dostupné z: <https://is.muni.cz/th/ghcai/>
- [17] SEONG, Boon. *What Is PIN1, PIN2, PUK1 & PUK2* [online]. Dostupné z: <https://almost-a-technocrat.blogspot.cz/2010/08/what-is-pin1-pin2-puk1-puk2.html>
- [18] ANDROIDCENTRAL. What is an APN, and how do I change it? 20.3.2018 [online]. [vid. 2018-03-05]. Dostupné z: <https://www.androidcentral.com/what-apn-and-how-do-i-change-it>
- [19] DIGI INTERNATIONAL. *Digi Connect Application Guide Cellular IP Connections* [online]. 2005 [vid. 2018-03-20]. Dostupné z: http://ftp1.digi.com/support/documentation/appguide_connectcellular_ipconsiderations.pdf
- [20] GOOGLE. *Google Git* [online]. Dostupné z: <https://android.googlesource.com/platform/frameworks/opt/mms/+4bfcd8501f09763c10255442c2b48fad0c796baa/src/java/com/google/android/mms/pdu>
- [21] W3C. *Synchronized Multimedia Integration Language* [online]. [vid. 2018-03-15]. Dostupné z: <https://www.w3.org/TR/SMIL2/>
- [22] GOOGLE. *Android source codes* [online]. Dostupné z: <https://android.googlesource.com/platform/frameworks/opt/mms/+4bfcd8501f09763c10255442c2b48fad0c796baa/src/java/com/google/android/mms/pdu>
- [23] GOOGLE. *Android Developers* [online]. 2017 [vid. 2018-03-05]. Dostupné z: <https://developer.android.com/index.html>

- [24] JAKE KLINKER. *klinker41/android-smsmms*. *GitHub* [online]. 2017 [vid. 2017-11-21]. Dostupné z: <https://github.com/klinker41/android-smsmms>
- [25] GOOGLE. *Google SubscriptionManager* [online]. Dostupné z: <https://developer.android.com/reference/android/telephony/SubscriptionManager.html>
- [26] WHISPERSYSTEMS. *Signal-Android* [online]. 2017. Dostupné z: <https://github.com/WhisperSystems/Signal-Android>
- [27] TLDRLEGAL. *GNU General Public License v3* [online]. Dostupné z: [https://tldrlegal.com/license/gnu-general-public-license-v3-\(gpl-3\)](https://tldrlegal.com/license/gnu-general-public-license-v3-(gpl-3))
- [28] WHISPERSYSTEMS. *JobManager*. *GitHub* [online]. 2017 [vid. 2017-11-21]. Dostupné z: <https://github.com/WhisperSystems/jobmanager>
- [29] FIREBASE. *Firebase Documentation* [online]. [vid. 2018-03-22]. Dostupné z: <https://firebase.google.com/docs>

Přílohy

1. Obsah přiloženého DVD
 - a. Zdrojové kódy aplikace
 - b. Instalační soubor aplikace (.apk)
2. Veřejný repositář v systému BitBucket s poslední verzí aplikace
 - a. <https://bitbucket.org/domestos99/mmsparams/src/Signal>
3. Aplikace na Google Play
 - a. <https://play.google.com/store/apps/details?id=cz.uhk.bak.mmsparams>

Oskenované zadání práce

Univerzita Hradec Králové
Fakulta informatiky a managementu
Akademický rok: 2017/2018

Studijní program: Aplikovaná informatika
Forma: Prezenční
Obor/komb.: Aplikovaná informatika (ai3-p)

Podklad pro zadání BAKALÁŘSKÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Plašil Dominik	Mírová 1807/61, Velké Meziříčí	I1500413

TÉMA ČESKY:

Mobilní aplikace pro podporu testování MMS centra

TÉMA ANGLICKY:

Mobile Application Supporting Testing of MMS Center

VEDOUCÍ PRÁCE:

Ing. Pavel Kříž, Ph.D. - KIKM

ZÁSADY PRO VYPRACOVÁNÍ:

Cíl: Vytvořit mobilní aplikaci pro operační systém Android. Aplikace bude implementovat standardní funkcionalitu MMS klienta a to komunikaci s MMS centrem (odesílání a příjem zpráv) rozšířenou o analýzu a možnost nastavení široké škály parametrů popsaných ve specifikaci 3GPP TS 23.140.

Osnova:

1. Úvod
2. Cíl
3. Rešerše existujících řešení
4. SMS, MMS a související technologie
5. Android API pro práci s SMS a MMS
6. Analýza a návrh vlastního řešení
7. Implementace
8. Výsledky
9. Závěr

SEZNAM DOPORUČENÉ LITERATURY:

- 1) LE BODIC, Gwenael. Mobile messaging technologies and services: SMS, EMS, and MMS. 2nd ed. Hoboken, NJ: J. Wiley, c2005. ISBN 0470011432.
- 2) <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=794>
- 3) <https://developer.android.com>
- 4) <https://github.com/klinker41/android-smsmms>

Podpis studenta: Plašil Dominik

Datum: 4.10.17

Podpis vedoucího práce: Pavel Kříž

Datum: 4.10.17