

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií



Diplomová práce

**Vývoj aplikace pro měření a vizualizaci propustnosti
bezdrátových sítí**

Roman Fürst

© 2018 ČZU v Praze

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Roman Fürst

Informatika

Název práce

Vývoj aplikace pro měření a vizualizaci propustnosti bezdrátových sítí

Název anglicky

Implementation measuring and visualizing wireless networks application

Cíle práce

Diplomová práce je tematicky zaměřena na problematiku sběru dat o bezdrátových sítích v okolí.

Hlavním cílem diplomové práce je navrhnout mobilní aplikaci, která bude měřit a perzistovat informace o bezdrátových sítích vzhledem k aktuálnímu času a poloze zařízení, a naměřená data vhodně interpretovat uživateli.

Díličí cíle diplomové práce jsou:

- vytvoření kritické literární rešerše k problematice vývoje mobilních aplikací a WiFi sítí,
- navrhnout a implementovat mobilní aplikaci,
- otestovat aplikaci v reálném prostředí,
- syntetizovat výsledky práce a formulovat závěry práce.

Metodika

Metodika řešené problematiky diplomové práce vychází ze studia a analýzy odborných informačních zdrojů. Praktická část práce je zaměřena na implementaci mobilní aplikace která měří, ukládá a vyhodnocuje informace o bezdrátových sítích v okolí. Na základě syntézy teoretických poznatků a výsledků praktické části práce budou formulovány závěry diplomové práce a bude stanoven návrh pro další rozšíření aplikace a využití nashromážděných dat.

Doporučený rozsah práce

60-80

Klíčová slova

mobilní aplikace, android, bezdrátové sítě, indoor navigace, měření, propustnost

Doporučené zdroje informací

- Ana Barroso, Matthias Hollick. Performance evaluation of delay-tolerant wireless friend-to-friend networks for undetectable communication. USA, Florida: Local Computer Networks (LCN), 2015. ISBN 978-1-4673-6770-7
- Mohib Shah, Jinman Kim, Dagan Feng. Impact of wireless devices over real-time applications: An empirical test-bed analysis. Sydney: 38th Annual IEEE Conference on Local Computer Networks – Workshops, 2013. ISBN 978-1-4799-0540-9
- Muhammad Faheem Awan, Tahir Ahmad, Saad Qaisar, Nick Feamster, Srikanth Sundaresan. Measuring broadband access network performance in Pakistan: A comparative study. USA, Florida: Local Computer Networks Conference Workshops, 2015. ISBN 978-1-4673-6773-8
- Mu Zho, Qiao Zhan, Yu Wang. Hotspot Ranking Based Indoor Mapping and Mobility Analysis Using Crowdsourced Wi-Fi Signal View Document. IEEE, 2017. ISSN 2169-3536
- Rajeev Shorey, A. Ananda, Mun Choon Chan, Wei Tsang Ooi. Measuring Wireless LANs. Wiley-IEEE Press, 2005. ISBN 9780471755593
- Yeong-Sheng Chen, Der-Jiunn Deng, Chun-Chien Teng. Range-Based Localization Algorithm for Next Generation Wireless Networks Using Radical Centers. IEEE, 2016. ISSN 2169-3536

Předběžný termín obhajoby

2017/18 LS – PEF

Vedoucí práce

Ing. Jan Tyrychtr, Ph.D.

Garantující pracoviště

Katedra informačních technologií

Elektronicky schváleno dne 30. 10. 2017

Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 1. 11. 2017

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 29. 03. 2018

Čestné prohlášení

Prohlašuji, že svou diplomovou práci " Vývoj aplikace pro měření a vizualizaci propustnosti bezdrátových sítí" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 31.3.2018

Poděkování

Rád bych touto cestou poděkoval Ing. Janu Tyrychtrovi Ph.D. za odborné vedení a pomoc při zpracování diplomové práce.

Vývoj aplikace pro měření a vizualizaci propustnosti bezdrátových sítí

Abstrakt

Diplomová práce se zabývá problematikou vývoje mobilní aplikace, která měří a zobrazuje propustnost bezdrátových sítí. Teoretická část práce je zaměřena na témata vývoje mobilních aplikací a vlastností bezdrátových sítí. Vypracovaná literární rešerše se věnuje definici mobilních aplikací, jejich členění, a postupům při vývoji na odlišných mobilních platformách. Dále jsou uvedeny a popsány vlastnosti bezdrátových sítí, především pak Wi-Fi sítí.

Praktická část práce se zaměřuje na návrh a implementaci mobilní aplikace. Jsou stanoveny požadavky na aplikaci, provedena analýza stavu konkurence, a za pomoci jazyka UML vymodelován návrh chování aplikace. Součástí teoretické části je také slovní popis implementace klíčových částí aplikace doplněný o grafické ukázky.

Výstupem práce je funkční prototyp mobilní aplikace pro platformu Android, která umožňuje dlouhodobě měřit a následně vizualizovat propustnost vybraných bezdrátových sítí. Funkčnost aplikace je otestována v reálných podmínkách a výsledky testu jsou následně vyhodnoceny v závěru práci.

Klíčová slova: mobilní aplikace, android, bezdrátové sítě, indoor navigace, měření, propustnost

Implementation measuring and visualizing wireless networks application

Abstract

This thesis summarizes issues during mobile application development for measuring and visualization wireless networks. The theoretical part is focused mainly on mobile applications development issues and wireless networks attributes. There are definitions and dividing of mobile applications, and develop technique on different mobile platforms stated in critical literary research. There are also stated and described parameters of wireless networks, especially Wi-Fi.

The practical part is focused on design and implementation of mobile application. There are summarized applications requirements, competitive analysis, and due UML language there are application behaviour described. There are also described key part of implementation in practical part include graphic illustrations.

The output is functional Android platform application prototype, which is able to measure and visualize selected wireless networks. In the next step application is tested in real environment and results are summarized in thesis conclusion.

Keywords: mobile applications, android, wireless networks, indoor navigation, measuring, throughput

Obsah

1 Úvod.....	10
2 Cíl práce a metodika	11
2.1 Cíl práce	11
2.2 Metodika	11
2.2.1 Softwarová analýza.....	12
2.2.2 Návrh	12
2.2.3 Implementace.....	13
2.2.4 Testování.....	14
3 Teoretická východiska	15
3.1 Úvod do problematiky vývoje mobilních aplikací.....	15
3.1.1 Kategorizace	15
3.1.2 Charakteristiky.....	18
3.2 Metodika vývoje mobilních aplikací.....	22
3.2.1 Vývojový proces	22
3.2.2 Platformy	28
3.3 Problematika Wi-Fi sítí.....	32
3.3.1 Bezdrátové sítě obecně	32
3.3.2 IEEE 802.11	35
3.3.3 Wi-Fi sítě	37
4 Vlastní práce	45
4.1 Softwarová analýza	45
4.1.1 Vize	45
4.1.2 Srovnání konkurence	45
4.1.3 Požadavky.....	48
4.2 Návrh.....	49
4.2.1 Výběr platformy.....	49
4.2.2 Ukládání dat.....	50
4.2.3 Vícevláknová architektura	50
4.2.4 Diagram případů užití	51
4.2.5 Diagram aktivit	51
4.2.6 Stavový diagram	54
4.2.7 Sekvenční diagram.....	54
4.3 Implementace	58
4.3.1 Obecný přístup.....	58
4.3.2 Postup implementace.....	59

4.4	Testování	64
4.4.1	Testovací scénář měření propustnosti.....	64
4.4.2	Testovací scénář zobrazení propustnosti	65
5	Výsledky	66
6	Závěr.....	70
7	Seznam použitých zdrojů	71
	Seznam obrázků	74
	Seznam tabulek	75

1 Úvod

Pro měření propustnosti sítě lze najít několik praktických důvodů. Jedním z nich může být například potřeba větších firem zmapovat interní bezdrátovou síť, detekovat slabé články a zjistit vytížení sítě v čase. Na základě vytěžování a vyhodnocování sesbíraných dat pak lze upravit topologii sítě, změnit konfiguraci síťových prvků nebo investice do výkonnějších síťových zařízení. Měření propustnosti může mít také význam více vědecký, kdy se například může zkoumat vliv okolního prostředí a jiných exogenních proměnných na propustnost sítě.

Měření propustnosti sítě lze využít i k porovnání hardwarových zařízení například o různé výkonnosti nebo od různého výrobce. Pokud bude měření probíhat ve stejný čas a za podobných, nebo ideálně stejných, podmínek můžeme při vhodné interpretaci porovnávat výkon jednotlivých prvků.

V neposlední řadě by měřená propustnost sítě mohla posloužit při řešení problému navigace uvnitř budov. Při vnitřní navigaci (anglicky Indoor Navigation) neurčuje mobilní zařízení svou polohu pomocí GPS souřadnic ale za pomoci vyhodnocení dat přijatých z okolních síťových vysílačů v dosahu. V první fázi, kdy skládá mapa pokrytí se tak může pro tyto účely hodit nejen měření signálu ale i propustnosti. Následně se další zařízení touto mapou řídí a odhadují tak na základě porovnání aktuálních dat a dat z mapy svou polohu.

Pro svou mobilitu, a používání bezdrátového síťového adaptéru se jako ideální prvek pro měření propustnosti nabízejí mobilní zařízení, a s tím spojená možnost osvojit si metodiku vývoje mobilní aplikace pro konkrétní platformu a provést implementaci vlastní aplikace, která bude splňovat definované požadavky.

2 Cíl práce a metodika

2.1 Cíl práce

Hlavním cílem diplomové práce je navrhnout mobilní aplikaci, která bude měřit a perzistovat informace o bezdrátových sítích vzhledem k aktuálnímu času a poloze zařízení, a naměřená data vhodně interpretovat uživateli.

Dílní cíle diplomové práce potom jsou:

- Vytvoření kritické literární rešerše k problematice vývoje mobilních aplikací a Wi-Fi sítí
- Navrhnout a implementovat mobilní aplikaci
- Otestovat aplikaci v reálném prostředí
- Syntetizovat výsledky práce a formulovat závěry práce

2.2 Metodika

Metodika řešené problematiky v této práci vyhází ze studia a analýzy odborných informačních zdrojů, především se zaměřením na témata mobilních aplikací, bezdrátových sítí a témata s nimi úzce související. Na základě informací získaných z analýzy odborné literatury byla na tato témata vytvořena literární rešerše potřebná k pochopení problematiky a správnému postupu při řešení teoretické části práce. Jelikož se jedná o téma úzce spojené s informačními technologiemi, čerpání informací probíhalo z aktuálních zdrojů především online v elektronické podobě. Hledání vědeckých článků a publikací probíhalo z velké části na vědeckých databázích jako je Scopus, Wiley, Web of Science, SpringerLink, ScienceDirect.

Navazující praktická část je zaměřena především na vytvoření funkčního prototypu mobilní aplikace, která bude schopna v delším časovém úseku měřit propustnost vybraných bezdrátových sítí, data vhodně perzistovat a následně je graficky zobrazit

uživateli. V následujících odstavcích je detailněji popsán postup při vypracování vlastní práce.

2.2.1 Softwarová analýza

V první části byla provedena softwarová analýza. Na začátku byla definována vize, která v několika větách definuje cíl a účel implementace nové mobilní aplikace. Následně byla provedena analýza trhu, a srovnání konkurenčních mobilních aplikací nabízející podobnou funkcionalitu jako aplikace vyvíjena v této práci. Na základě analýzy bylo zjištěno, že se na současném trhu nachází mezera v nabídce aplikací pro měření propustnosti Wi-Fi sítí. Jelikož byly nalezeny pouze dvě funkční aplikace, které splňovaly požadavky, upustilo se od plánované vícekritériální analýzy variant, která by v tomto případě poztrácela význam. Z analýzy tedy vyplynulo, že současná situace nahrává k implementaci a poskytnutí vlastního řešení problematiky měření propustnosti bezdrátových sítí.

S přihlédnutím na informace načerpané z literární rešerše a softwarové analýzy bylo rozhodnuto, že jako cílová platforma, pro kterou se bude aplikace v rámci této práce vyvíjet, bude Android OS. Bylo přihlédnuto nejen k tomu, že Android je v současné době nejpopulárnější mobilní operační systém běžící na drtivě většině mobilních zařízení, ale také fakt, že zařízení pro Android platformu jsou mnohem lépe dostupná než pro kteroukoliv jinou mobilní platformu. Tento fakt hrál velkou roli, jelikož pro vývoj a následné testování aplikace v reálných podmínkách je vlastnictví mobilního zařízení nezbytnost.

2.2.2 Návrh

Po softwarové analýze následoval návrh aplikace. Pomocí grafického jazyka UML (Lano, 2009) byl namodelován diagram aktivit, stavový diagram, sekvenční diagram a use-case diagram. Zatímco diagram aktivit zobrazuje průchod aktivitami napříč celou aplikací, tak stavový diagram popisuje stavy klíčové aktivity, která provádí opakované měření. Sekvenční diagram popisuje interakci uživatele a jednotlivými aktivitami a vláken aplikace

s operačním systémem Android v čase. Diagram případů užití zachycuje skutečnost, že aplikace může být uživatelem použita pro měření nebo zobrazení propustnosti.

2.2.3 Implementace

Po dokončení návrhu započala samotná implementace aplikace. Vývoj probíhal v oficiálním vývojovém prostředí Android Studio v objektově orientovaném programovacím jazyce Java. Při používání interních tříd a knihoven platformy Android byly informace čerpány z oficiálních webových stránek (developer.android.com), kde se nachází veškerá dokumentace pro konkrétní verzi operačního systému, včetně praktických ukázek použití.

Jelikož aplikace nepoužívá výpočtově náročné operace, ale naopak operace, které jsou náročné spíše časově, byl využit princip více vláknového programování (Cavalheiro, Pilla, 2014). Bylo tím dosaženo toho, že se maximálně využila výpočetní kapacita zařízení a zároveň se zkrátil celkový čas prováděných operací v daném okamžiku.

Pro zobrazení dostupných sítí v okolí bylo vytvořeno vlastní ListView. Pomocí třídy WifiManager která pracuje s Wi-Fi adaptérem zařízení, bylo provedeno oskenování okolí, a informace o sítích byly prezentovány uživateli v podobě naplněného ListView. Zároveň byl zaregistrován OnWifiChangeListener, který při jakékoliv změně Wi-Fi okolí provedené nové skenování, a aktualizuje seznam dostupných sítí.

Jedním z nefunkčních požadavků byl, aby aplikace byla schopna dlouhodobě běžet na pozadí operačního systému. Pro tento účel nemohly být použity standartní Android aktivity, ale využily se takzvané Servisy (angl. Services), které jsou schopny běžet i v případě že samotná aplikace není momentálně na popředí. V servisním vlákně běží čítač, který po vypršení času spustí asynchronně nové měření. Dále bylo potřeba zajistit, aby operační systém po delší době neaktivity nevyvídal nadbytečné hardwarové prvky, z důvodu úspory energie (například Wi-Fi adaptér). Toho bylo dosaženo tím, že byl zaregistrován takzvaný WakeUpBroadcastReceiver, který v krátkém časovém intervalu vysílá budící impuls, a udržuje zařízení stále v aktivním stavu.

Řešení klíčového požadavku měření propustnosti bylo řešeno tak, že aplikace si stahuje v definovaném časovém intervalu konkrétní datový soubor o konstantní velikost uložený v interní paměti zařízení. Bylo využito faktu, že současné Wi-Fi adaptéry jsou polo duplexní, a bylo možné dosáhnout toho, že aplikace (tedy mobilní zařízení) odesílá data sama sobě a zároveň nedochází k rušení signálu. Aplikace ve dvou výpočetních vláknech běží současně jako server i jako klient na jednom zařízení. Propustnost je následně spočítána jako podíl přeneseného objemu dat za jednotku času (Lo, 2007). Vyslaný paket tedy proteče přes přístupový bod (AP) do síťového směrovače (router) a zase zpět.

Perzistování dat nebylo řešené pomocí SQL lite, kterou Android nabízí, ale pomocí jednoduchého csv souboru který obsahuje jednotlivé záznamy po řádcích. Výhoda csv souborů je ta, že je snadno přenositelný a je široce podporovaný ve statistických, analytických a vědeckých programech pro zpracování dat. Aplikace ukládá nejen časový otisk a úroveň naměřené propustnosti, ale i další informace, která se později mohou hodit při vytěžování dat (síla signálu, MAC adresa síťového prvku, odezva, frekvence).

Pro vizualizaci naměřených dat v podobě grafu byla použita externí knihovna GraphView která byla pro účely této práce nepatrně upravena, aby správně interpretovala data nasbírána v řádu několika dní. Ke grafu byla za pomoci vlastního ListView připojena legenda, která nabízí uživateli základní informace k sítím zobrazených v grafu. Ke každé křivce v grafu je uveden název sítě, začátek a konec měření sítě, maximální a minimální naměřená propustnost, frekvence.

2.2.4 Testování

Funkčnost aplikace byla otestována v reálném prostředí. V průběhu několika dnů byla měřena stejná síť ve zhruba stejném časovém úseku během dne. Mobilní zařízení bylo umístěno na stejném místě, aby měření bylo co nejobjektivnější. Výstupem měření bylo pět csv souborů. Po načtení souborů aplikací byl výstupem graf se několika křivkami interpretující průběh propustnosti sítě v každém s měřených dnů.

3 Teoretická východiska

3.1 Úvod do problematiky vývoje mobilních aplikací

V posledních několika letech byl zaznamenán prudký nárůst trendu mobilních zařízení. V první řadě se jedná o chytré telefony a tablety. S tím je spojena i rostoucí poptávka po mobilních aplikacích. Každým rokem se počet vyvinutých, stažených a používaných aplikací rapidně zvyšuje. Díky tomuto rostoucímu trendu tak v podstatě vniklo nové odvětví a prostor pro podnikání jak softwarovým firmám, tak jednotlivým vývojářům. V mnoha ohledech je vývoj mobilních aplikací stejný jak například u standartních desktopových aplikací, ale přesto s sebou vývoj mobilního softwaru nese jistá specifika. Díky širokému spektru platforem, zařízení a vývojových nástrojů je pro úspěšný vývoj úspěšné aplikace potřeba tato specifika pochopit a vhodným způsobem zvolit následnou metodiku vývoje. Následující kapitola shrnuje specifické vlastnosti mobilních aplikací a doporučení pro jejich vývoj.

3.1.1 Kategorizace

Mobilní aplikace jak celek můžeme rozčlenit do několika typů a kategorií na základě funkcionality a zaměření. Pro úspěšný návrh a vývoj je potřeba pochopit rozdíl mezi mobilními a tradičními aplikacemi. Každá mobilní aplikace, která je dnes dostupná, by se dala začlenit zařadit do správně kategorie v následujících tabulkách:

Tabulka 1: Typy mobilních aplikací

	Typ	Popis
1	Webová	Aplikace není nainstalovaná na zařízení. Je dostupná přes webový prohlížeč na konkrétní url. Data nejsou uložena přímo v paměti zařízení. (příklad google.com)

2	Nativní	Je nainstalována na konkrétním zařízení. Její funkcionality nevyžaduje žádný přenos dat na server a zpět. Aplikace může fungovat bez síťové připojení a data jsou uložena v paměti zařízení, (příklad kalendář, poznámky)
3	Hybridní (web)	Aplikace je sice nainstalována na konkrétním zařízení ale pro svou činnost vyžaduje permanentní síťové připojení (příklad Facebook, Skype)
4	Hybridní (mix)	Aplikace je nainstalovaná na zařízení a ke své činnosti může i nemusí vyžadovat síťové připojení. Využívá offline data a v momentě navázání konektivity dojde k online aktualizaci hodnot.

Zdroj: FLORA, WANG, SWATI, (2014)

Rozhodnutí zda se bude jednat o mobilní aplikaci nativní nebo webovou je třeba rozhodnout již v počátcích projektu, ve fázi návrhu. Následující seznam uvádí případy kdy je optimální volba nativní aplikace (Mcwherter, Gowell. 2012) :

- nároky na grafický a výpočetní výkon
- využití fotočipu zařízení
- využití mikrofonu zařízení
- práce s adresářovou strukturou
- používání notifikací
- využití servis běžících na pozadí
- jedná se o hru

Jestliže předchozí tabulka nastiňovala rozdělení aplikací dle typu, následující tabulka a ukazuje

dělení dle kategorií:

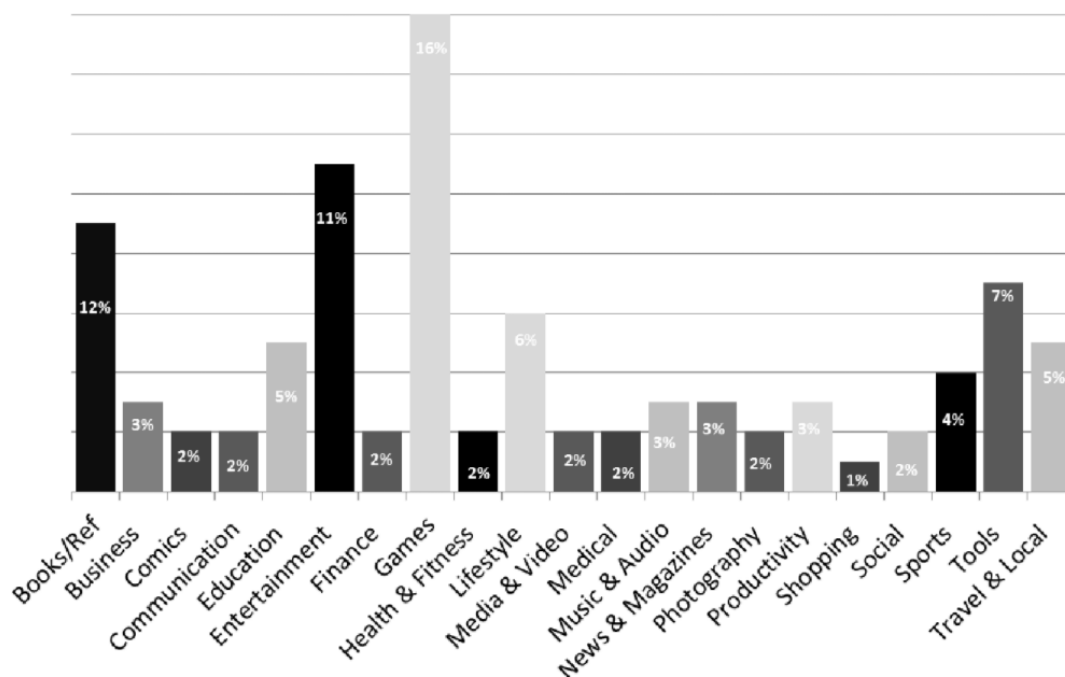
Tabulka 2: Kategorie mobilních aplikací

	Kategorie	Popis
1	Komunikace	Emailový klient, sociální sítě, internetový prohlížeč, zpravodajské portály, atd...
2	Hry	Online/offline, sportovní, akční, logické, atd...
3	Multimédia	Prohlížeč fotek, videí, prezentací, audio přehrávače, streaming, atd...
4	Produktivita	Kalendář, úkolovník, kancelářské nástroje, bankovní aplikace, záznam hovoru, atd...
5	Cestování	Městský průvodce, převodník měn, GPS navigace, překladač, počasí, atd...
6	Nástroje	Správce souborů, správce obrazovky, správce hovorů, atd...
7	Vzdělání	Početní nástroje, tabulky, kurzy, atd...

Zdroj: FLORA, WANG, SWATI, (2014)

Následující obrázek ukazuje detailnější kategorizaci mobilních aplikací pro platformu Android, společně s grafickou ukázkou rozdělení aplikací. Data jsou sesbíraná ze statistik Google Play (dříve Android Market):

Obrázek 1: Kategorizace a procentuální zastoupení aplikací pro platformu Android



Zdroj: MCWHERTER, GOWEL (2012)

Z obrázku patrné, že mobilní aplikace lze členit do řady kategorií dle jejich zaměření. I když jedna z charakteristik mobilních aplikací je ta, že by měly být jednoúčelové, není to vždy pravidlem a aplikace svou funkcionalitou může spadat do několika kategorií zároveň. Její zařazení pak není jednoznačné. Dle hodnot z obrázku lze však jasně určit že nejčastěji jsou mobilní aplikace využívány pro hry a zábavu. Na druhou stranu lze vidět, že například aplikace s vědeckým zaměřením v grafu vůbec zastoupeny nejsou.

3.1.2 Charakteristiky

Mobilní aplikace mají největší výhodu oproti standartním programům v tom, že se dají využívat na mnoha místech a nejdou vázány na pevnou pozici jako v případě desktop aplikací. Nabízejí relativně velký výpočetní výkon a relativně malé rozměry. Obdobně jako při vývoji tradičních programů, i u mobilních aplikací se při návrhu a vývoji setkáváme se základními problémy jako například hardwarová integrace, výkon, spolehlivost, omezení datové prostoru nebo třeba bezpečnostní opatření. Je však celá řada požadavků, kterými se

vývoj mobilních aplikací liší od tradičního přístupu. Mobilní aplikace totiž často kombinují svou funkčnost s funkčností dalších aplikací, pracují se sensory zařízení, musí pracovat na mnoha hardwarových a softwarových platformách, řeší bezpečnost přenosu dat, přizpůsobování uživatelského rozhraní. V neposlední řadě se při návrhu mobilního softwaru řeší také nároky na spotřebu energie a je klade důraz na agilní přístup vývoje s krátkými iteracemi. Charakteristiky kvality mobilních aplikací definuje standart ISO 9126. Standart strukturuje definici softwaru do následující kategorii: funkčnost – spolehlivost – použitelnost – efektivnost – spravovatelnost - přenosnost (anglicky functionality – reliability – usability – efficiency - maintainability - portability). (Moumane, Abran, 2016)

Další podstatnou charakteristickou vlastností, kterou bychom měli brát v potaz a patřičně s ní naložit je fakt, že mobilní zařízení jsou prakticky osobní počítače. Na rozdíl od pevných stanic jsou mnohem více vázány na konkrétního uživatele a na jeho osobní potřeby. Stejně tak mobilní aplikace může mnohem více pracovat s osobními údaji uživatele. Na základě vyhodnocování dat, predikce chování a potřeb uživatele pak aplikace může v průběhu času přizpůsobovat svou funkčnost.

Na základě dat z odborných anket, ve kterých byli dotazováni jak uživatelé, tak odborníci zabývající se vývojem mobilních aplikací lze shrnout a definovat obecné charakteristiky mobilních aplikací. V anketách se objevovaly dotazy týkající se jak užívání a prožitku z užívání mobilní aplikace tak dotazy spojené s omezením během vývoje. Vybrané charakteristiky byly rozděleny do dvou částí: hardwarové a softwarové. Hlavní charakteristiky mobilních aplikací jsou uvedeny v následujících odstavcích (Flood, Duce, 2013):

Hardwarové charakteristiky

Spotřeba zdrojů

Mobilní zařízení čerpají energii z akumulátoru. Zároveň mají oproti stolím standartním počítačům nižší výkon procesu a paměť. Aplikace určená pro mobilní platformy by s těmito aspekty měla počítat a být navržena tak aby zbytečně neplýtvala omezenými zdroji mobilního zařízení. Aplikace a její zdrojový kód by měl být co nejstručnější a měl by obsahovat je potřebnou funkcionalita

Vstupy

U mobilních aplikací je interakce s uživatelem velice důležité. Stejně tak způsob jakým uživatel zadává aplikaci potřebné vstupy. Jelikož většina moderních chytrých zařízení není vybaveno hardwarovou klávesnicí a tlačítky, měl by být počet potřebných textových vstupů omezen a vstup by měly být řešeny kliknutím, a výběrem předdefinované možnosti na obrazovce aplikace.

Rozlišení

Velikost, rozlišení a tvar obrazovky mobilních zařízení je podstatným faktorem pro návrh aplikace. Je třeba mít na mysli, že aplikace bude běžet na celé řadě zařízeních s odlišnou velikostí a rozlišením displeje. Tomu by mělo odpovídat tak rozložení, velikost a množství prvků na obrazovce.

Spuštění

Mobilní aplikace jsou používány odlišně než v případě standartních aplikací. K jejich využití dochází několikrát denně a s relativně krátkým časem, po který je aplikace aktivní. Je zapotřebí, aby čas načtení aplikace byl co nejkratší.

Fragmentace

Mobilní aplikace mohou běžet na mnoha hardwarových platformách, softwarových platformách a zároveň na několika verzích konkrétní platformy.

Softwarové charakteristiky

Uživatelský prožitek, uživatelské rozhraní

Interakce mezi aplikací a uživatelem je v případě mobilních aplikací klíčová. Grafické rozhraní by mělo být pro uživatele příjemné a dopřát mu co nejlepší prožitek. Aplikace by měla nabídnout bezproblémové použití funkcionality a zároveň uživatele přinutit použít aplikaci znovu v budoucnu. Rozhraní by mělo být přehledné a intuitivní, responsivní, a je třeba mít na paměti, že aplikace může běžet na několika typech přístrojů a rozložení a volba grafických prvků by tomu mělo odpovídat.

Interakce

Ať už se jedná nativní, nebo online aplikaci, vždy je zapotřebí aby aplikace byla schopna spolehlivě komunikovat s okolními zdroji. Může se jednat například o systémové prostředky operačního systému, databázi daného zařízení, kamerovým nebo GPS senzorem, nebo interakci s jinými aplikacemi pomocí internetové sítě, V každém případě by taková interakce měla být spolehlivá, rychlá a flexibilní a neměl by omezit uživatelské použití aplikace.

Lokalizace

Souvisí úzce s předchozím bodem interakce. Jedná se ale o konkrétní použití GPS adaptéru integrované přímo v zařízení, a využití aktuální polohy k úpravě chování nebo obsahu aplikace. Opět se jedná o jednu z předních výhod oproti standartním desktopovým aplikacím

Zaměření

Zaměření mobilních aplikací je na rozdíl od klasických programů více specifické a jednoúčelové. Aplikace by měla nabízet konkrétní specifickou funkčnost, a její použití by mělo být co nejjednodušší a nejpřímější.

Bezpečnost

Pokud aplikace pracuje s citlivými daty nebo je přenáší přes internetovou síť, měla by s daty zacházet bezpečně například za použití šifrování a bezpečnostních transportních

protokolů. Data by měla být zabezpečena v takové míře, aby nemohlo dojít k vědomému útoku během přenosu, nebo ukládání, dat. Zároveň by aplikace měla být na tolik chytrá, aby byla schopná validovat vstupní uživatelská data a zaručovala integritu dat (například zadání hodnot mimo rozsah, nebo kliknutí dvakrát na tlačítko odeslat, atd...)

3.2 Metodika vývoje mobilních aplikací

Metodika vývoje mobilních aplikací je z velké části shodná s metodikou vývoje běžných aplikací. Přesto se však tradiční postupy vývoje nedají plně aplikovat i v případě mobilních aplikací. V souladu se specifickými požadavky na mobilní aplikace a zařízení je třeba mírně upustit klasického postupu a zvolit agilnější přístup k řešení problematiky (Harleen, 2014). Následující kapitola se snaží nastínit a shrnout řešení na následující témata: jaké jsou současně metodologické postupy při vývoji mobilních aplikací. Na jaké problémy a nedostatky můžeme v průběhu vývoje narazit. Jaká jsou praktická doporučení pro vývoj mobilních aplikací.

V druhé části je shrnut přehled v současnosti nejpoužívanější mobilních platform a jejich stručná charakteristika.

3.2.1 Vývojový proces

V návaznosti na odlišnou specifikaci mobilních aplikací je doporučeno v praxi k vývoji přistupovat odlišně než v případě desktopového nebo serverového softwaru. V současné době nepokrývá téma metodiky a přístupu vývoje žádný mezinárodní standart, ale bylo na toto téma vypracováno několik studií, a každá k problematice přistupuje lehce odlišně. Například Leigh Williamson definuje čtyři hlavní prvky - způsob uživatelského vstupu, použitelnost, design a volba vhodné implementační technologie pro konkrétní typ aplikace. Naproti tomu Anthony Wasserman klade důraz na vývojové procesy, nástroje, návrh uživatelského rozhraní, přenosnost, kvalitu a bezpečnost aplikace. Josh Dehlinger zase vyzdvihuje univerzálnost uživatelského rozhraní, použitelnost skrze platformy, návrh

aplikace pracující s kontextem a rovnováhu mezi agilním a tradičním vývojem. (Flora, 2014)

Vývoj mobilních aplikací je relativně mladé informační odvětví a je často spojováno s agilní metodikou. Tradiční (vodopádový) přístup, tedy analýza-design-implementation-testování-údržba není v tomto prostředí běžné. Vývoj probíhá spíše agilně. Běžná je například metodika SCRUM nebo Test Driven Development Velkou výhodou agilního přístupu je hlavně komunikace v týmu kdy se každý může zapojit na návrhu aplikace a způsobu řešení požadavků (Curcio, Navarro, Tmalucelli, 2018). U vývoje mobilních aplikací je také běžná praxe že se prvotně sesbírají požadavky od klienta, a následně se sestaví prototyp aplikace a proběhne prezentace nefunkčního prototypu. Následně se upřesní požadavky, definuje se plán, harmonogram a milníky, a následně započne samotná implementace.

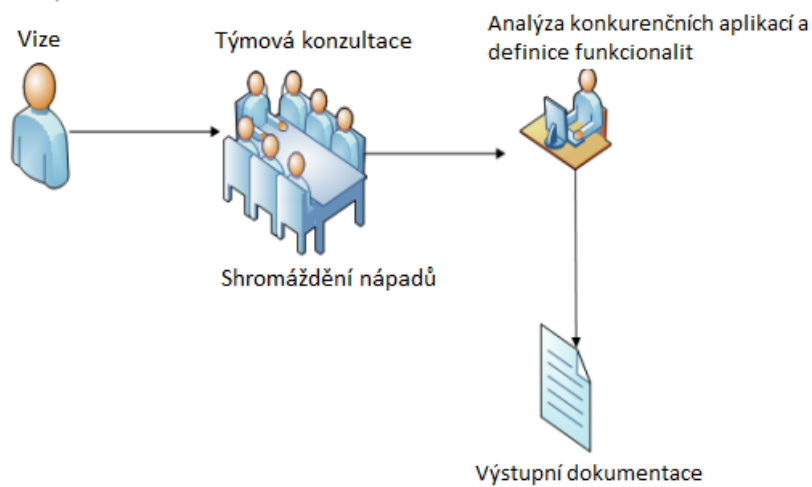
Celý proces vývoje mobilních aplikací je spíše zaměřen na dílčí iterace, ve kterých je dodávána a testována určitá část funkcionality. Po otestování a posouzení klientem / zadavatelem se pokračuje vývojem další verze prototypu a implementuje se další část funkčnosti.

Je potřeba mít na paměti, že neexistuje jednotný postup vývoje softwaru. Záleží na mnoha aspektech jako například prostředí vývoje, cílová skupina uživatelů, finanční prostředky, hardwarové prostředky, typ aplikace, typ zpracovatelské firmy, časové omezení atd., Přesto bychom mohli vývojový proces shrnout do následujících fází:

Identifikace

V této úvodní fázi probíhá sběr požadavků a analýza. Klient/zadavatel je součástí schůzek a definuje svou vizi. Analyzuje se současný stav nabízených aplikací s obdobnou funkcionalitou napříč platformami a navrhuje se inovativní řešení pro novou aplikaci. Upřesňuje se definice a cíle celého projektu společně s požadavky. Výstup této fáze by měl být zdokumentován a jasně popsán.

Obrázek 2: Fáze identifikace

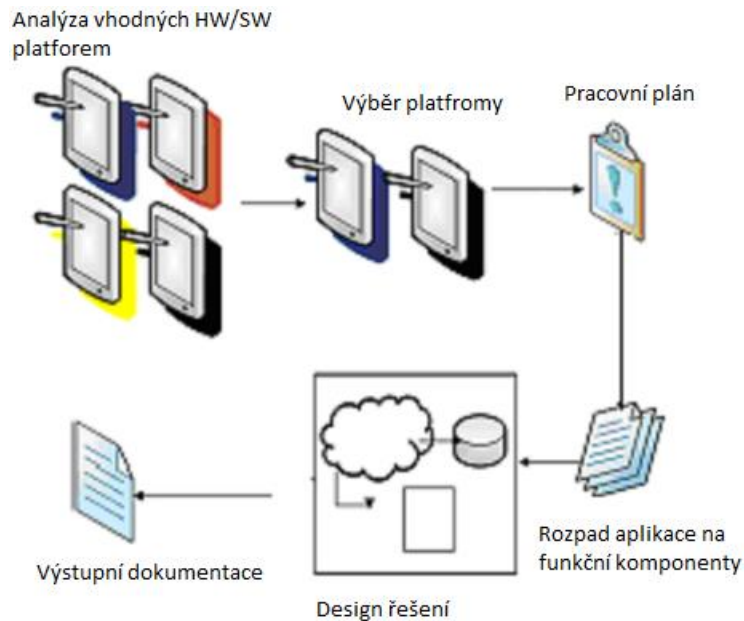


Zdroj: VITHANI, KUMAR. (2014)

Design

Na základě identifikace se stanoví platforma, na které následně bude probíhat implementace aplikace. Rozhoduje se, zda aplikace bude placena nebo volně dostupná ke stažení. Probíhá rozpad funkcionality aplikace na jednotlivé moduly a části. Definuje se rozhraní, architektura aplikace, popřípadě definice datové struktury. Výstupem je kompletní technická dokumentace, která je základním prvkem pro vývoj. Je důležité, aby se dokumentace udržovala kompletní a aktuální.

Obrázek 3: Fáze designu



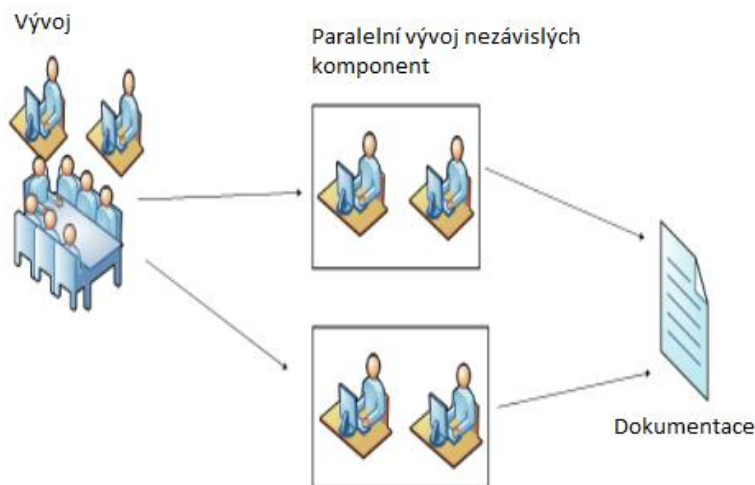
Zdroj: VITHANI, KUMAR. (2014)

Vývoj

Na základě návrhu probíhá implementace prototypu aplikace. Nejdříve se implementuje jádro aplikace a stěžejní funkcionality, a posléze se přidávají okolní prvky. Pokud je aplikace rozdělena do několika samostatných komponent, jednotlivé komponenty stejného prototypu mohou být implementovány paralelně. Stejně tak může probíhat paralelní vývoj funkcionality a grafiky uživatelského rozhraní. Následně je proveden integrace jednotlivých komponent do jednoho funkčního prototypu.

Jestliže, je aplikace cílena na více platforem, je doporučeno, aby aplikace byla pro každou platformu vyvíjena totožně a vzhled aplikace byl u každé platformy pokud možno stejný. Vývoj pro jednotlivé platformy může probíhat taktéž paralelně. Výstupem vývojové fáze je funkční prototyp aplikace a dokumentace vývoje.

Obrázek 4: Vývojová fáze

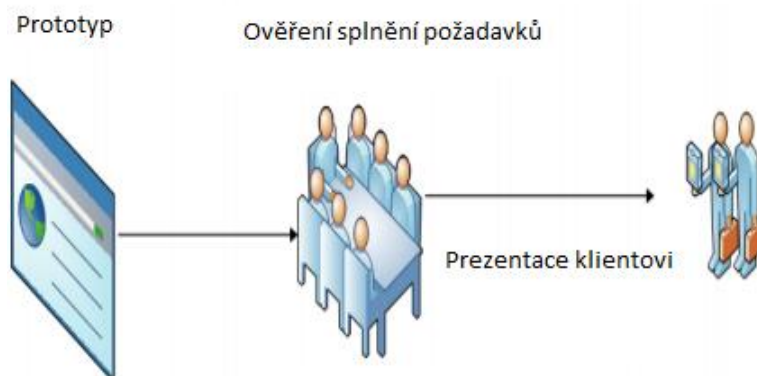


Zdroj: VITHANI, KUMAR. (2014)

Prototypování

Prototypování je velice častým prvkem mobilního vývoje. Výstupem vývojové fáze je funkční prototyp, následně je ověřena jeho funkčnost, a prezentace zákazníkovi, který reaguje zpětnou vazbou. Po schválení se pokračuje vývojem další verze prototypu navazující na verzi předchozí. Celý cyklus prototypování a vývoje se opakuje, dokud není vyvinuta finální verze funkčního prototypu.

Obrázek 5: Fáze prototypování



Zdroj: VITHANI, KUMAR. (2014)

Testování

Finální fází procesu vývoje je testování. Testuje se nejenom finální verze aplikace, ale testování probíhá v průběhu celého vývoje. Testovat by se mělo na všech cílových platformách, na odlišných verzích operačního systému a na různých typech zařízení. V závislosti na rozsahu a komplexnosti aplikaci se průběžně provádějí jednotkové testy, integrační testy, regresní testy, zátěžové testy nebo může být na určitou dobu zaveden takzvaný pilotní provoz, kdy je aplikace využívána v reálném prostředí úzkým okruhem uživatelů

Obrázek 6: Fáze testování



Zdroj: VITHANI, KUMAR. (2014)

Základem úspěchu vývoje a samotné aplikace je plánování. Před započítím práce bychom měli mít jasno, za jakým účelem chceme aplikaci vyvíjet a jaký má přínos oproti konkurenci, která je v současnosti na trhu. Dalším velice důležitým krokem je správná a úplná definice požadavků. Čím dříve a přesněji se nám podaří definovat všechny prvky a požadavky která tím spíše se chytne nesrovnalostem a chybám v pokročilých fázích projektu. V případě mobilních aplikací je také třeba věnovat pečlivost nejen ve fázi návrhu architektury aplikace a volby platformy, ale v neposlední řadě se zaměřit na návrh uživatelského rozhraní a zaměřit se aby aplikace nabídla uživateli co nejlepší uživatelský prožitek. (Flora, 2014)

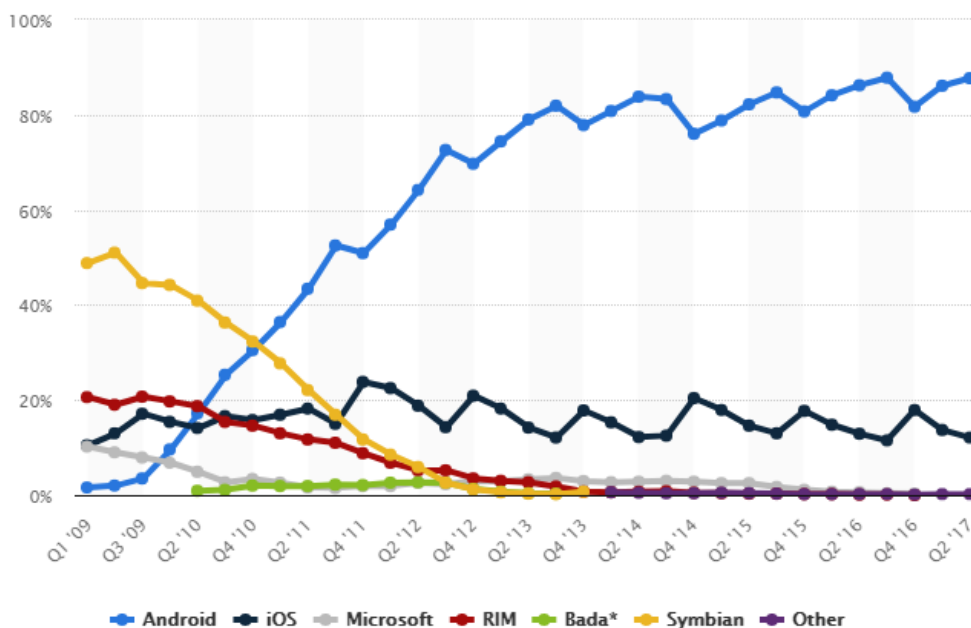
3.2.2 Platformy

Pokud se bavíme mobilních aplikacích, v aktuálně je na výběr několik platforem, a to nejen softwarových, ale hlavně hardwarových. Například v současnosti nejrozšířenější mobilní platforma Android je k dispozici hned v několika verzích, na několika typech (smart tv, navigace, chytré telefony, tablety, ...) zařízení od několika značek po celém světě (Sony, Samsung, Huawei, ...). Při návrhu aplikace hraje výběr platformy velikou roli, protože každá z platforem nabízí určitá specifika a s nimi spojen určitý okruh koncových uživatelů a specifikuje využití aplikace.

Pokud se zaměříme na softwarové platformy, v současně době má smysl vyvíjet aplikace na třech nejčastějších mobilních platformách. Android, iOS a Microsoft Mobile. (Gartner 2015).

Na následujícím obrázku ze statistického portálu Statista je ukázán graf který vystihuje trend procentuálního použití jednotlivých platforem. Z grafu je jasně čitelné, že platforma Android je již v několika letech prudce na vzestupu. Na druhém místě je s velkým odstupem iOS který si řadu drží svůj standart.

Obrázek 7: Vývoj zastoupení mobilních platforem na trhu (2017)



Zdroj: www.statista.com

Android

Na trhu od roku 2003, v současné době vlastněn a vyvíjen společností Google. Jedna se se nejrozšířenější platformu, která distribuována jako open-source, díky čemuž trpí značnou fragmentací. V současnosti se vyskytuje v několika verzích na nespočtu typech mobilních zařízení. Díky svému širokému záběru Android není, a ani nemůže, být plně optimalizován na všech zařízeních, a v některých případech může docházet až k nekompatibilitě mezi softwaru. Tento fakt je platformě Android často vyčítán. Hlavně na slabších zařízeních se můžou novější verze OS Android jevit jako těžkopádné. (Zikmund, 2015)

Nejaktuálnější dohledatelná data o přehledu četnosti jednotlivých instalací verzí operačního systému Android byla dohledána na oficiálních stránkách platformy, a je z nich patrné, že v současné době jsou všechny instalace operačního systému Android fragmentovány do 7 verzí. Tři poslední verze, tedy Nougat (verze 7), Marshmallow (verze 6) a Lollipop (verze 5) jsou zastoupeny nejpočetněji se zhruba stejným procentuálním podílem.

Architektura Androidu je postavena na Linuxovém jádře, nad kterým jsou postaveny nativní C/C++ knihovny a běhové prostředí (ART). ART je virtuální prostředí pro aplikace napsané v programovacím jazyce Java. Na rozdíl od Java Virtual Machine (JVM) běžících na desktopových platformách, je ART optimalizované pro běh na mobilních zařízeních.

Vývojové prostředí

Google aktuálně nabízí pro vývoj aplikací na své platformě vývojové prostředí Android Studio založeném na produktu IntelliJ IDEA. Jelikož vývoj Android aplikací probíhá v jazyce Java, Android Studio obsahuje kompletní SDK pro bezproblémový vývoj. Běhové prostředí pro Java není postaveno na Java Virtual Machine (JMV) jako je tomu u desktopů, ale na virtuální stroji Android Runtime (ART). (developer.android.com 2018).

iOS

Pod názvem iPhone OS uveden na trh v roce 2007. Platforma není distribuována jako open-source a je úzce vázaná na zařízení od značky Apple. Distribuce operačního systému je vysoce marketingově řízena a kontrolována a nenabízí takovou volnost, jako je tomu například u Android OS.

Z dat poskytnutých na oficiálních webových stránkách společnosti Apple lze vyčíst, že ze všech instalací operačního systému iOS je nejpočetněji (65%) zastoupena nejnovější verze iOS 11. S velkým odstupem je pak iOS verze 10 (28%) a zbytek doplňují starší verze.

Architektura je postavena na kernelu Mach. Nad jádrem je několik vrstev zajišťující interakci s hardwarem, a jako poslední vrstva je právě Framework Cocoa Touch, který umožňuje vývoj aplikací. (developer.apple.com 2018)

Vývojové prostředí

Pro vývoj aplikací na iOS je dostupný jediný vývojový nástroj IDE Xcode které je použito pro všechny platformy od společnosti Apple. Xcode využívá programovací framework Cocoa, který je postaven na jazyku Objective-C . Díky použitému frameworku Cocoa je zajištěno, že výsledný vzhled aplikace bude splňovat směrnice společnosti Apple.

Windows Phone

Operační systém Windows Phone vydala Společnost Microsoft v roce 2010 jako nástupce Windows Mobile. Systém není volně distribuován a společnost striktně určuje licenční podmínky a parametry zařízení. Oproti systému Android nebo iOS jsou Windows Phone velice řídně zastoupeny a jsou na ústupu. Koncem roku 2017 společnost Microsoft oznámila, že v budoucnu s podporou platformy již nepočítá.

Na základě informací poskytnutých webovým portálem Adduplex, který je oficiálními stránkami pro Windows Phone a Windows Store, lze uvést, že Windows Phone se vyskytují v současné době ve čtyřech verzích. Ze všech instalací toho mobilního operačního systému je jeho nejnovější verze 10 zastoupena v 17%. Předchozí verze 8.1 se vykytuje na 76,63% všech zařízeních se systémem Windows Phone.

3.3 Problematika Wi-Fi sítí

3.3.1 Bezdrátové sítě obecně

Bezdrátové sítě se dělí dvou základních skupin. První skupinou jsou mobilní sítě, kam patří například 2G, 3G, 4G, 5G sítě. Druhá skupina jsou sítě spadající do standardu IEEE 802 (Bellalta, 2016). Sem spadá například Bluetooth, Wi-Fi, WiMax. Všechny typy bezdrátových sítí můžeme také rozdělit na základě rozlohy, kterou síť pokrývá, a s ní spojený způsob využití. Nejčastější je dělení na PAN, LAN, MAN, WAN. Následující tabulka nabízí stručné srovnání bezdrátových sítí, kde je pro daný typ sítě uvedeno orientační pokrytí, výkon, příklad standardů spadajících do dané kategorie, a příklad použití daného typu sítě:

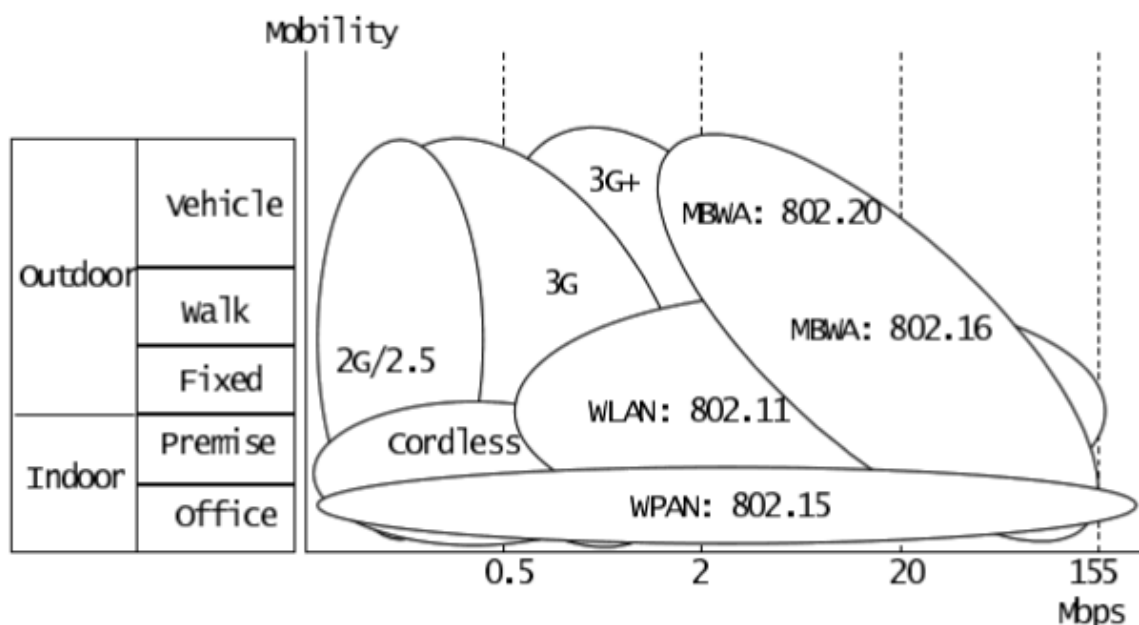
Tabulka 3: Srovnání bezdrátových sítí

Typ	Pokrytí	Výkon	Standarty	Použití
WPAN	V dosahu osob	Střední	Bluetooth, IEEE 802.15, IrDa	Nahrazení kabeláže k periferiím
WLAN	Budovy, korporace	Vysoký	IEEE 802.11, Wi-Fi, HiperLAN	Nahrazení lokálních metalických sítí
WMAN	Města, metropole	Vysoký	IEEE 802.16, WiMax	Připojení celých domácností a firem k internetu
WAN	Celosvětově	Nízký	2G, 3G, 4G	Mobilní přístup k Internetu odkudkoliv

Zdroj: KANIKA (2015)

Následující obrázek nastiňuje vztah jednotlivých typů sítí v závislosti na mobilitě a přenosové rychlosti:

Obrázek 8: Přehled bezdrátových sítí v závislosti na mobilitě a přenosové rychlosti



Zdroj: PRASAD, PRASAD (2005)

Z obrázku je patrný rozdíl mezi bezdrátovými mobilními sítěmi a sítěmi ze standardu IEEE 802. Například WLAN sítě sice nabízejí vysokou rychlost ale jejich mobilita je značně omezena. Jsou vázány na konkrétní místo a pokrývají jen přilehlé okolí. Naproti tomu mobilní 3G sítě nabízejí středně vysokou rychlost ale jejich mobilita je mnohem vyšší a připojení je možné provozovat prakticky odkudkoliv.

Bezdrátové sítě fungují na principu šíření signálu pomocí radiových vln. V závislosti na typu konkrétní bezdrátové sítě se volí anténa vysílače tak, aby odpovídala charakteristice sítě. V zásadě rozdělujeme antény na:

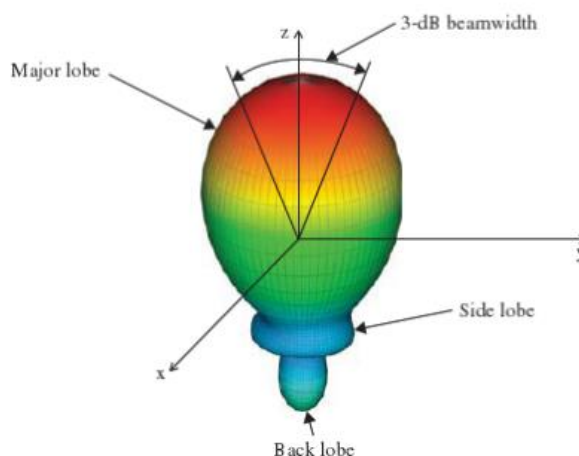
Směrové antény

Jsou použity na větší vzdálenost. Například pro vzájemné propojení dvou LAN sítí. Vysílaný paprsek signálu je delší ale o to plošší. Záleží tedy, jak je anténa nasměrována a jakým směrem je paprsek vysílán (Kanika, 2015).

Pro představu si lze její využití představit při pokrytí signálem dlouhou chodbu. Pokrytí však bude při použití směrové antény pouze v prvním patře. V patrech nad nebo pod nikoliv.

Tvar signálu ze směrové antény připomíná žárovku:

Obrázek 9: Vizualizace signálu ze směrové antény

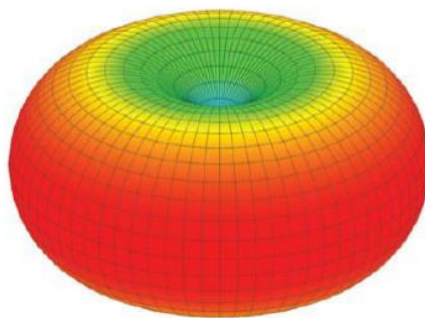


Zdroj: (GAO, LUO, ZHU, 2013)

Všesměrové antény

Použití na ucelené pokrytí menšího prostoru. Paprsek není tak dlouhý ale je vysílán do všech stran. Slouží například ke kompletnímu pokrytí kanceláře. (Kanika, 2015). Tvar signálu ze všesměrové antény připomíná tvar koblíhy:

Obrázek 10: Vizualizace ze všesměrové antény



Zdroj: (GAO, LUO, ZHU, 2013)

Z obrázků je jasně vidět rozdíl v šíření signálu pomocí směrové a všesměrové antény. Zatím co u směrové antény se signál šíří jedním směrem, všesměrové antény signál pokrývá celé okolí vysílače. Barvy pak znázorňují intenzitu signálu. Modrá barva, která se vykytuje u zdroje vysílání, indikuje vysokou úroveň signálu, zatímco červená, která je od vysílače vzdálená, reprezentuje nízkou úroveň signálu.

3.3.2 IEEE 802.11

Celosvětově uznávaný standard, který definuje lokální bezdrátové sítě (WLAN). Standard 802.11 se zaměřuje především na dvě nízké úrovně vrstvy ISO/OSI modelu – na spojovou vrstvu (MAC) a na fyzickou vrstvu (PHY). Standard se od roku 1997 postupně vyvíjel a nyní existuje ve více jak 22 různých verzích (Shuang, Biju, 2015). Různé verze standardu používají různé modulace signálu. Nejčastěji se jedná o DSSS, FHSS, OFDM a jejich kombinace. Modulace je podrobněji zmíněna v pozdější části práce. Stručný přehled jednotlivých verzí je uvede v následující tabulce (Lo, 2007), (Holík, 2014):

Tabulka 4: Přehled verzí standardu IEEE 802.11

Verze	Rok	Frekvence	Přenosová max. rychlost	Modulace
IEEE 802.11	1997	2,4GHz	1-2 Mbit/s	FHSS, DSSS
IEEE 802.11b	1999	2,4 GHz	5,5-11 Mbps	FHSS, DSSS
IEEE 802.11a	1999	5GHz	54 Mbit/s	OFDM
IEEE 802.11j	1999	2,4 GHz	54 Mbit/s	OFDM
IEEE 802.11g	2003	2,4 GHz	52 Mbit/s	ERP-DSSS, ERP-OFDM
IEEE 802.11y	2008	3,7 GHz	54 Mbit/s	
IEEE 802.11n	2009	2,4GHz, 5GHz	600 Mbit/s	MIMO OFDM
IEEE 802.11ac	2014	2,4GHz, 5GHz	1000 Mbit/s	MU-MIMO OFDM

Zdroj: vlastní

Nejnovější verze bezdrátových sítí IEEE 802.11ac nabízejí díky šířce pásma 160MHz a vysoké modulaci 256QAM teoretickou propustnost mezi stanicemi až 1 Gbit/s . Za použití technologie multiuser MIMO (multiple input – multiple output), využívající k přenosu více antén najednou na více frekvencích, se 802.11ac sítě svým chováním přibližují plně duplexním sítím.

3.3.3 Wi-Fi sítě

Wi-Fi sítě spadají do standardu IEEE 802.11 a jedná se tedy o bezdrátové LAN sítě (WLAN). Wi-Fi sítě pracují především na frekvenci 2,4GHz nebo 5GHz, a šířka pásma je stanovena mezi 20 – 25 MHz (Bellalta, 2016),

Fyzikální veličiny

Frekvence

Počet oscilací signálu za jednu vteřinu. Frekvence f se udává v hertzech (Hz)

Vlnová délka

Vzdálenost, kterou urazí jedna vlna za dobu jednoho frekvenčního kmitu. Vlnová délka λ se v případě bezdrátových sítí uvádí v metrech.

$$\lambda = \frac{c}{f} [m]$$

Kde c je rychlost světla [m/s] a f je frekvence [Hz].

Vlnová délka se u Wi-Fi sítí s frekvencí 2,4 GHz pohybuje okolo 12,5cm, a u frekvence 5 GHz okolo 6cm.

Šířka pásma

Šířka pásma fyzické vrstvy lze definovat jako spektrální šířku elektromagnetických signálů – rozdíl minima a maxima v daném frekvenčním pásmu (Prasad, Prasad 2005)

Fáze

Časový posun mezi dvěma signály. Jednotkou je stupeň (°)

Amplituda

Fyzikální veličina značící maximální odchylku od střední veličiny s pravidelným opakováním. U bezdrátových amplituda definuje úroveň signálu. Signál se uvádí v decibelech [dB].

Zvyšování amplitudy se nazývá zisk, snižování ztráta.

Modulace

Signál se šíří pomocí radiových vln. Pokud je ale signál vysílán na konstantní frekvenci a se stejnou fází a amplitudou, nenesou žádnou informaci. Aby signál přenášel informace, je zapotřebí aby se sinusovka signálu v čase měnila. Tento proces se nazývá modulace. (Dobkin, 2004).

Důvodem modulace způsob jakým interpretovat jednotlivé 1 a 0 v přenášeném signálu. Interpretace se provádí posunem nebo změnou frekvence od základní nosné frekvence. Na základě rozdílu změněné a základní frekvence se reprezentuje přechod stavu.

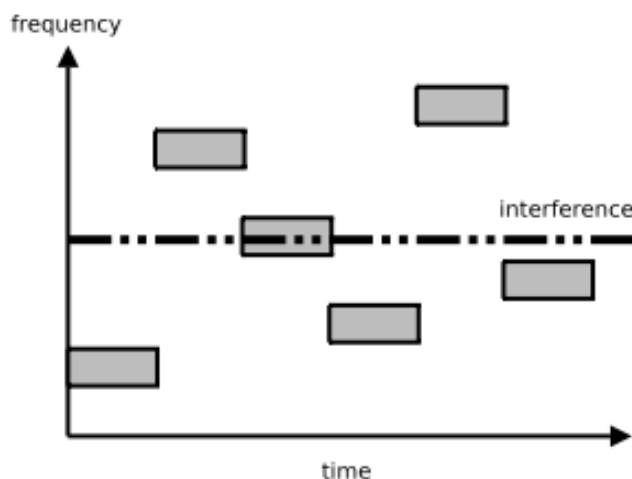
Pro modulaci se využívá princip rozprostřeného spektra, který definuje způsob jakým je přenášen radiový signál. Rozprostřené spektrum přenáší informace za pomoci matematické funkce rozložené do širokého frekvenčního intervalu. (Zandl, 2003)

V následujících odstavcích je uveden přehled technologií používaných k modulaci signálu Wi-Fi sítí.

Frequency-hopping spread spectrum – FHSS

Princip spočívá v přeskokování mezi frekvencemi. K jednotlivým přeskokům se využívá pásmo o šířce 83,5 MHz rozdělené na 79 kanálů. Při přenosu bitu vysílači přijímač pseudonáhodně přeskočí na jiný kanál. Různé dvojice vysílačů a přijímačů provádí své pseudonáhodné skoky jinak, aby se minimalizovala kolize na stejném kanálu.

Obrázek 11: Modulace FHSS



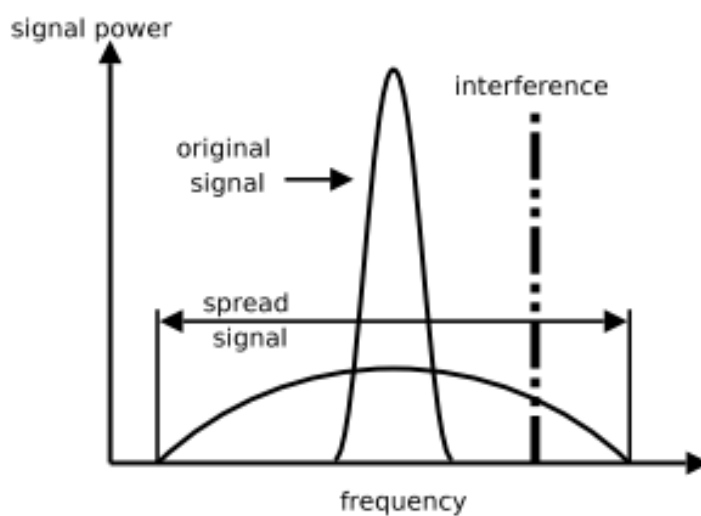
Zdroj: HOLT, HUANG, (2010)

Direct-sequence spread spectrum – DSSS

Signál je rozprostřen do většího radiového spektra. Využívá kanály šířce 22MHz což zajišťuje větší přenosovou rychlost a spolehlivost. (Holík, 2014.) Nejdříve je každý bit převeden na sekvenci pseudonáhodných bitů (angl. chip sequence), a výsledná sekvence se následně moduluje fázovým posunem.

Ukázka modulovaného signálu pomocí DSSS:

Obrázek 12: Modulace DSSS



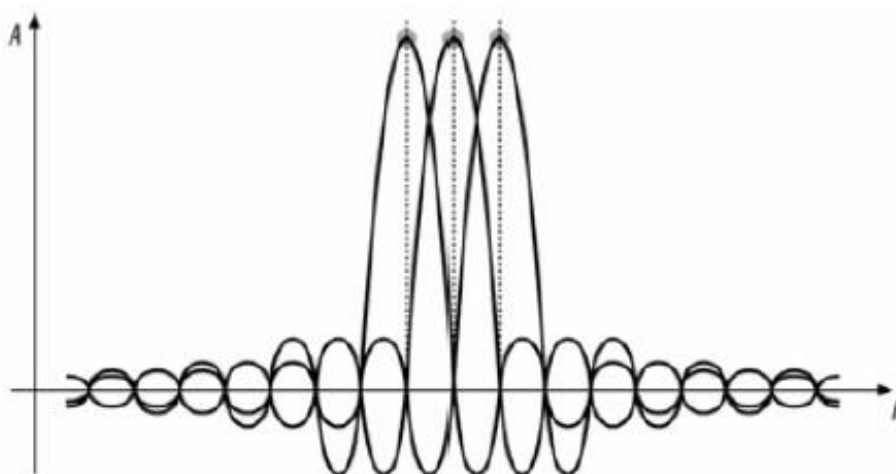
Zdroj: HOLT, HUANG (2010)

Orthogonal Frequency Division Multiplexing - OFDM

Rozděluje jeden široko-pásmový kanál na několik sub-kanálů, kterými se následně přenáší nezávislá data. Sub-kanály se sice překrývají, ale jsou uspořádány ortogonálně. Data jsou kódovaná na maximální amplitudě daného sub-kanálu – tam kde jsou amplitudy ostatních kanálů rovny nule. (Gast, 2005)

Na následujícím obrázku je zobrazeno uspořádání sub-kanálu. Tečky snaží maximální amplitudu:

Obrázek 13: Ortogonální uspořádání kanálu OFDM



Zdroj: GAST (2005)

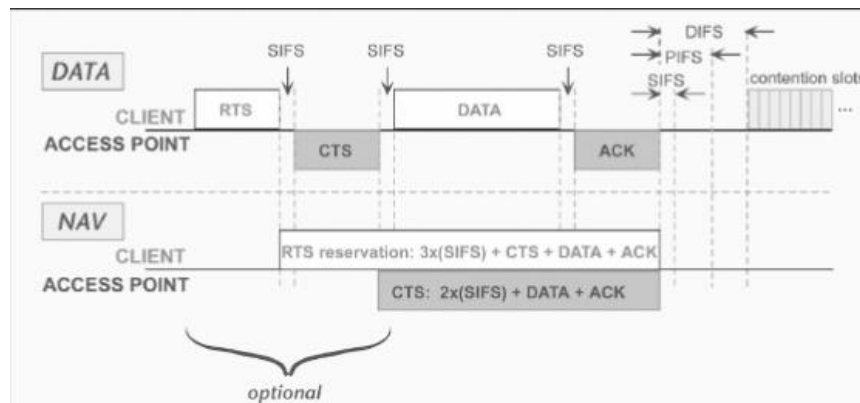
Přístup k médiu

Na rozdíl od metalických sítí, bezdrátové sítě ke komunikaci využívají sdílené medium a nejsou schopny detekovat kolize. Proto bezdrátové sítě fungují polo-duplexně. Tedy nemohou ve stejný okamžik vysílat a zároveň přijímat signál. Zároveň je zapotřebí ošetřit, aby nedocházelo ke kolizím mezi jednotlivými vysílači. Jinými slovy aby dvě zařízení nevysílala ve stejný okamžik. Z tohoto důvodu standart 802.11 definuje v rámci MAC vrstvy implementaci CSMA/CA (Carrier Sense Multiple Access / Collision Avoidance), která obstarává přístup jednotlivých prvků k mediu sítě, a minimalizuje tak riziko kolize. (Bellalta, 2016).

Jedním ze základních prvků jsou potvrzovací pakety (ACK). Po každém vysílání by měl vysílač od přijímače dostat potvrzovací paket. Pokud vysílač z nějakého důvodu ACK paket neobdržel (kolize, ztráta signálu, rušení), zvolí náhodné časové okno a vysílání zopakuje. Pro snížení četnosti ztracených ACK paketů z důvodu kolize existují dva hlavní přístupy implementace CSMA/CA. První přístup je za pomoc RTS paketů (Request To Send), který vyšle vysílač, a informuje tím ostatní, že chce vysílat paket o určité délce. Síťový prvek rezervuje požadovaný čas na odesílání dat plus jejich potvrzení, a informuje

vysílač CTS paketem (Clear To Send), čímž mu sdělí, že medium je rezervované pro jeho vysílání. Pro lepší pochopení, je tento přístup rezervace časového okna pomocí RTS, CTS a ACK paketů zobrazen na následujícím obrázku 14:

Obrázek 14: Výměna paketů při sdílení media



Zdroj: DOBKIN (2005)

Druhým přístupem je pak způsob, kdy se paket rozdělí na několik drobných fragment, které se následně odešlou. Větší počet menších fragmentů má větší šanci na bezproblémový průchod skrz síť, než jeden velký paket.

Parametry výkonnosti

Signál

Signál se v bezdrátových sítích vyjadřuje jako poměr mezi intenzitou signálu a šumu, respektive poměr dvou výkonů. Jelikož se u Wi-Fi sítí setkáváme s relativně malými výkony, vztahuje se výpočet na 1 miliwatt výkonu. Jednotka signálu je tedy decibel-miliwatt [dBm]. Vzorec pro výpočet signálu vypadá následovně (Lo, 2007. Holík 2014):

$$dBm = 10 * \log_{10} \frac{P_1}{1 mW}$$

Intenzita signálu se se také vyjadřuje ukazatelem přijímaného signálu – anglicky Received Signal Indicator (RSSI). Hodnota RSSI je bezrozměrná a relativní. Každé zařízení může tedy RSSI interpretovat rozdílně. Je prokázána lineární závislost mezi úrovní signálu a propustností (Prasad, 2005)

Přenosová rychlost

Rychlost, kterou je na fyzické vrstvě možné konstantně přenášet data bez ohledu na to o jaký typ dat se jedná. Přenosová rychlost je omezena vlastnostmi media. Jednotkou je Megabit za vteřinu [Mbps].

Propustnost

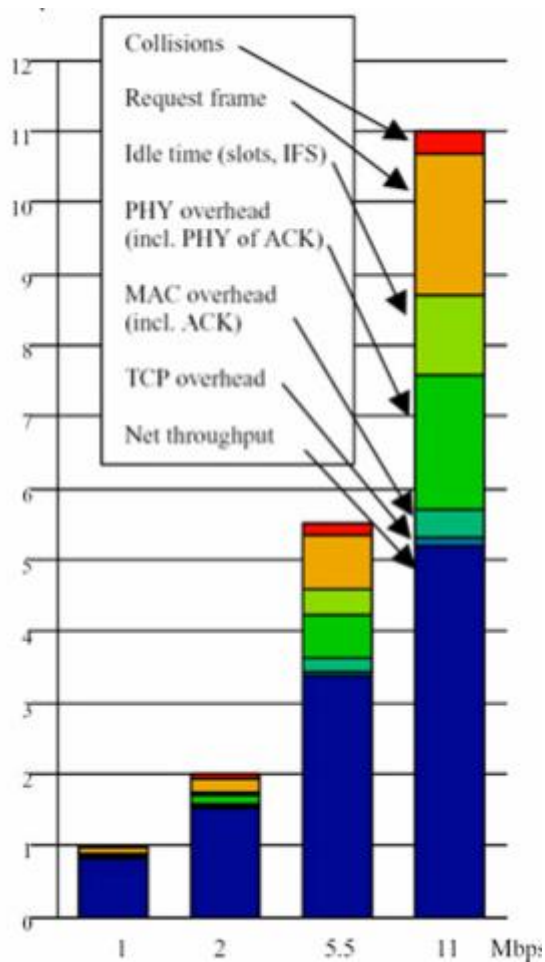
Propustnost sítě závisí na mnoha faktorech jako například na režii MAC protokolu, síle signálu, vzdálenosti mezi vysílačem a přijímačem, momentální provoz sítě, počtu kolizí, a další (Holík, 2014). Jde o rychlost toku dat v rámci sítě. Dá se tedy měřit jako podíl přenášených dat za jednotku času. Jednotka Megabit za sekundu [Mbps] (Dobkin, 2005).

Wi-Fi sítě jsou účinné zhruba ze 40% svého výkonu. Jinými slovy, pouze 40% celkového objemu dat jsou vlastní přenášené informace. Zbýlých 60% procent spolknou provozní činnosti. Potvrzovací pakety (35%), přeposílání ztracených paketů (15%), kolize a další režijní činnost (10%) (Lo, 2007). Dalším podstatným faktorem, který ovlivňuje celkovou propustnost sítě, je počet síťových prvků, kterými data prochází. Každý prvek navíc, který je na cestě mezi odesláním paketu a jeho dosažení

cíle, má za následek nárůst doby přenosu. Každý prvek v síti navíc nepatrně navyšuje přenosový čas, a tím snižuje propustnost sítě.

Následující obrázek zobrazuje využití propustnosti Wi-Fi sítí pro jednotlivé přenosové rychlosti. Modrá část grafu značí skutečnou propustnost sítě:

Obrázek 15: Skutečné využití celkové propustnosti Wi-Fi sítě



Zdroj: LO, 2007

4 Vlastní práce

Praktická část práce se zaměřuje na vývoj funkčního prototypu mobilní aplikace, která bude schopna měřit propustnost bezdrátových sítí. Obsahem této kapitoly je z velké části technická dokumentace, která modeluje a popisuje návrh struktury a chování aplikace.

4.1 Softwarová analýza

4.1.1 Vize

Vize, pro praktickou část této práce, je vytvořit mobilní aplikaci na vhodné mobilní platformě, která bude v definovaném časovém intervalu provádět měření propustnosti vybrané Wi-Fi sítě, nebo sítí. Měření by mělo probíhat v delším časovém horizontu (například celý den), a proto by aplikace měla být schopna běžet na pozadí, nebo při vypnutém displeji zařízení. Dále by aplikace měla data vhodně ukládat a později je interpretovat uživateli v podobě grafu. Graf by měl umět zobrazit několik křivek pro různé sítě najednou. Ke grafu by měla být připojena legenda s přehledným popisem jednotlivých zobrazených křivek. Aplikace by měla být co nejjednodušší bez zbytečných prvků a důraz by měl být kladen spíše na spolehlivost.

4.1.2 Srovnání konkurence

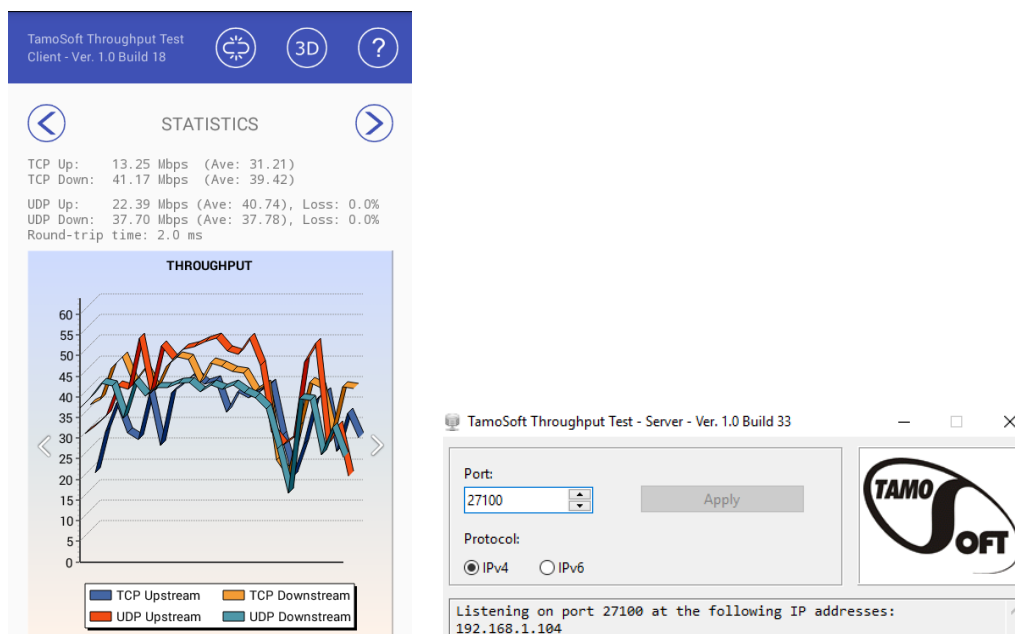
Byla provedena analýza aplikací, na základě které bylo zjištěno, že na trhu existuje celá řada mobilních aplikací, které nabízí analýzu Wi-Fi sítí. Většina těchto aplikací nabízí společnou funkcionalitu jako je například měření rychlosti připojení k internetu, měření rychlosti downloadu a uploadu, odezva, síla signálu. Společně s dalšími informacemi o síti jako například frekvence, IP adresa a šířka pásma, byla měřená data v reálném čase interpretována buď v podobě akcelerometru nebo grafů. Většina z nich však vyžaduje připojení k internetové síti. Tedy neměří propustnost v rámci lokální sítě ale propustnost připojení k vnější síti. Aplikací, které by se věnovaly skutečně

měření propustnosti v rámci lan sítě, bylo nalezeno doslova jen pár a v následujících odstavcích je jejich stručný přehled:

TamoSoft Throughput

Volně stažitelná aplikace. Její funkcionality je jednoúčelová. Pro svou funkčnost vyžaduje dvě zařízení, která jsou připojená ke stejné síti. Na jednom zařízení je nainstalovaná serverová část, a na druhém je klientská část. Měření propustnosti probíhá na stisk tlačítka. Naměřené hodnoty jsou zobrazeny ve formě lineárního grafu. Data nelze uložit pro pozdější použití. Aplikace nenabízí možnost dlouhodobého měření. Serverová i klientská část je dostupná pro Android, Windows, iOS a MacOS.

Obrázek 16: Ukázka aplikace TamoSoft Throughput

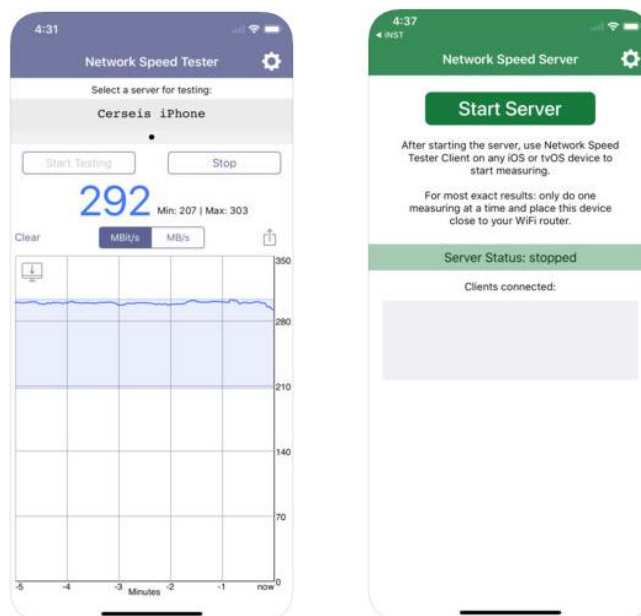


Zdroj: vlastní

Network Speed Tester

Jednoúčelová aplikace pro měření propustnosti sítě, která vyžaduje dvě zařízení, mezi kterými probíhá měření. Na jednom zařízení je nainstalována aplikace jako klient, a na druhém jako server. Klientská část je volně ke stažení pro platformy Android, iOS a MacOS. Serverová část je dostupná pouze pro iOS a MacOS a to pouze v placené verzi. Aby aplikace byla aktivní, musí běžet na popředí. Měření probíhá v reálném čase a data nelze nijak ukládat. Aplikace má jednoduchý a přehledný design, a její používání je intuitivní.

Obrázek 17: Ukázka aplikace Network Speed Tester



Zdroj: www.itunes.apple.com

Po provedení průzkumu a analýzy konkurence bylo zjištěno, že se na trhu nachází díra v oblasti měření propustnosti bezdrátových LAN sítí, kterou je možné vhodně využít a přijít s vlastním nápadem řešení. Jelikož bylo skutečně dohledáno minimum aplikací splňující požadavky, bylo upuštěno od multikriteriální analýzy, která by v tomto případě postrádala význam.

Klíčové vlastnosti současných mobilních aplikací pro měření propustnosti sítí, by se daly shrnout do následujících několika bodů:

- Jedná se o nativní aplikace
- Běží na dvou zařízeních v režimu klient nebo server
- Nenabízí uložení naměřených dat
- Nejsou schopny běžet na pozadí systému
- Jsou uživatelsky přívětivé
- Data interpretují v podobě lineárního grafu

4.1.3 Požadavky

V této kapitole jsou specifikovány požadavky na chování budoucí aplikace. Při definování požadavků bylo přihlédnuto k vizi a cílům vývoje, a k analýze trhu provedené v předchozí kapitole.

Nefunkční požadavky

- Nativní mobilní aplikace - aplikace musí být cílena na jednu z mobilních platforem
- Přehledné uživatelské rozhraní – důraz na příjemný uživatelský prožitek
- Intuitivní ovládání – jednotlivé kroky musí být jasně definované
- Stabilita chování – aplikace musí spolehlivě fungovat delší časový interval
- Přenositelnost dat – naměřená data mohou být exportována nebo importována
- Minimalistická – aplikace by měla spotřebovávat minimum systémových zdrojů

Funkční požadavky

- Měření propustnosti Wi-Fi sítě – měření propustnosti v rámci LAN sítě

- Zobrazení propustnosti – aplikace musí umět přehledně zobrazit naměřená data v podobě grafu
- Měření jedním zařízením – aplikace běží jako server i jako klient na jednom zařízení
- Automatické měření – měření se musí automaticky spouštět po uživatelem definovaném časovém intervalu
- Ruční měření – uživatel může v době mezi automatickým měření provést měření ručně
- Přenosnost naměřených dat – naměřená data by mělo být možné exportovat pro použití mimo aplikaci
- Perzistence dat - v průběhu měření musí být data ukládána

Vzhledem k tomu, že požadujeme, aby měření probíhalo za pomoci jednoho fyzického zařízení, je zřejmé, že propustnost bude měřená jen na úseku sítě mezi mobilním zařízením a prvním směrovacím prvkem sítě. Předpokládaná posloupnost fyzických prvků, kterými musí paket projít je: mobilní zařízení, přístupový bod, směrovač, přístupový bod, mobilní zařízení.

4.2 Návrh

Následující kapitola obsahuje technickou dokumentaci k návrhu mobilní aplikace. Pro modelaci struktury a chování aplikace v podobě diagramů byl použit grafický jazyk UML.

4.2.1 Výběr platformy

Jako cílová platforma byl zvolen Android OS. Hlavním důvodem je jeho rozšířenost. V současné době je Android nainstalován na drtivě většině mobilních zařízení a je používán většinou uživatelů. Cílová verze, pro kterou bude aplikace vyvíjena, je Oreo 8.0 (API 26) se zpětnou kompatibilitou až k verzi Lollipop 5.0 (API

21). Dalším důvodem výběru je i fakt, že zařízení s operačním systémem Android jsou mnohem dostupnější než zařízení s kteroukoliv jinou platformou. Jelikož pro vývoj a následné testování aplikace bylo nezbytné vlastnit alespoň jedno fyzické zařízení, byla volba OS Android pro účely této práce nejlepší variantou.

Zvolené vývojové prostředí, ve kterém bude probíhat implementace, je Android Studio ve verzi 2.3. Vývojové prostředí je volně ke stažení na oficiálních webových stránkách Android.

4.2.2 Ukládání dat

Operační systém Android nabízí relační databázi SQLite, která ovšem není vhodná pro konkrétní použití vyvíjené aplikace. Pro ukládání dat bylo zvoleno řešení, kdy se data ukládají do souboru formátu CSV. Formát dat v souboru je čitelný pro člověka i pro stroj. Výhoda toho řešení je, že se naměřená data dají snadno přenést do jiného softwaru, kde se s daty může dále pracovat. Každý záznam je na samostatném řádku a jednotlivé hodnoty odděleny středníkem. Každý řádek bude obsahovat údaje v následujícím pořadí: časový otisk, název sítě, naměřená propustnost, odezva, frekvence, maximální teoretická propustnost a MAC adresa přístupového bodu.

Soubory CSV (comma-separated valuse) jsou soubory obsahující tabulkově strukturovaná data. Každý řádek tabulky reprezentuje řádek v souboru, a sloupce tabulky jsou položky řádku oddělené čárkou nebo středníkem. V případě že položka v řádku obsahuje čárky, tečky nebo středníky, je celá hodnota uvedena do zdvojených uvozovek.

4.2.3 Vícevláknová architektura

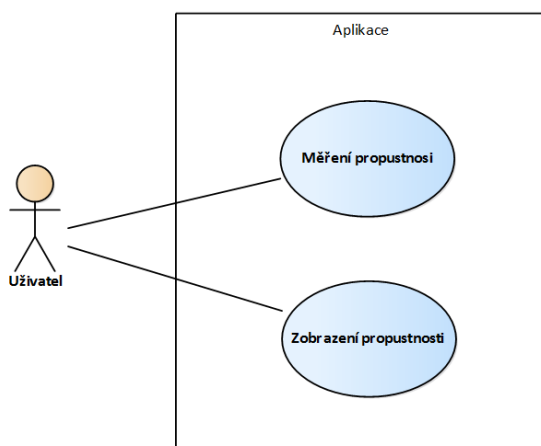
Programovací jazyk Java a platforma Android podporují vícevláknové programování. Využití více vláken se hodí například v případě provádění asynchronních operací, nebo případě kdy chceme minimalizovat čas vykonávání

několika operací s nízkou výpočetní náročností. Aplikace vyvíjená v této práci splňuje oba předpoklady. V souladu s požadavkem, aby aplikace byla schopna chovat se jako klient i jako server na jednom fyzickém zařízení, je využití principů vícevláknového programování nezbytností.

4.2.4 Diagram případů užití

Aplikace bude z hlediska možných případů užití jednoduchá. Bude nabízet jen dva způsoby použití: měření propustnosti a zobrazení propustnosti. Oba případy jsou zobrazeny v diagramu případů užití na následujícím obrázku:

Obrázek 18: Diagram případů užití



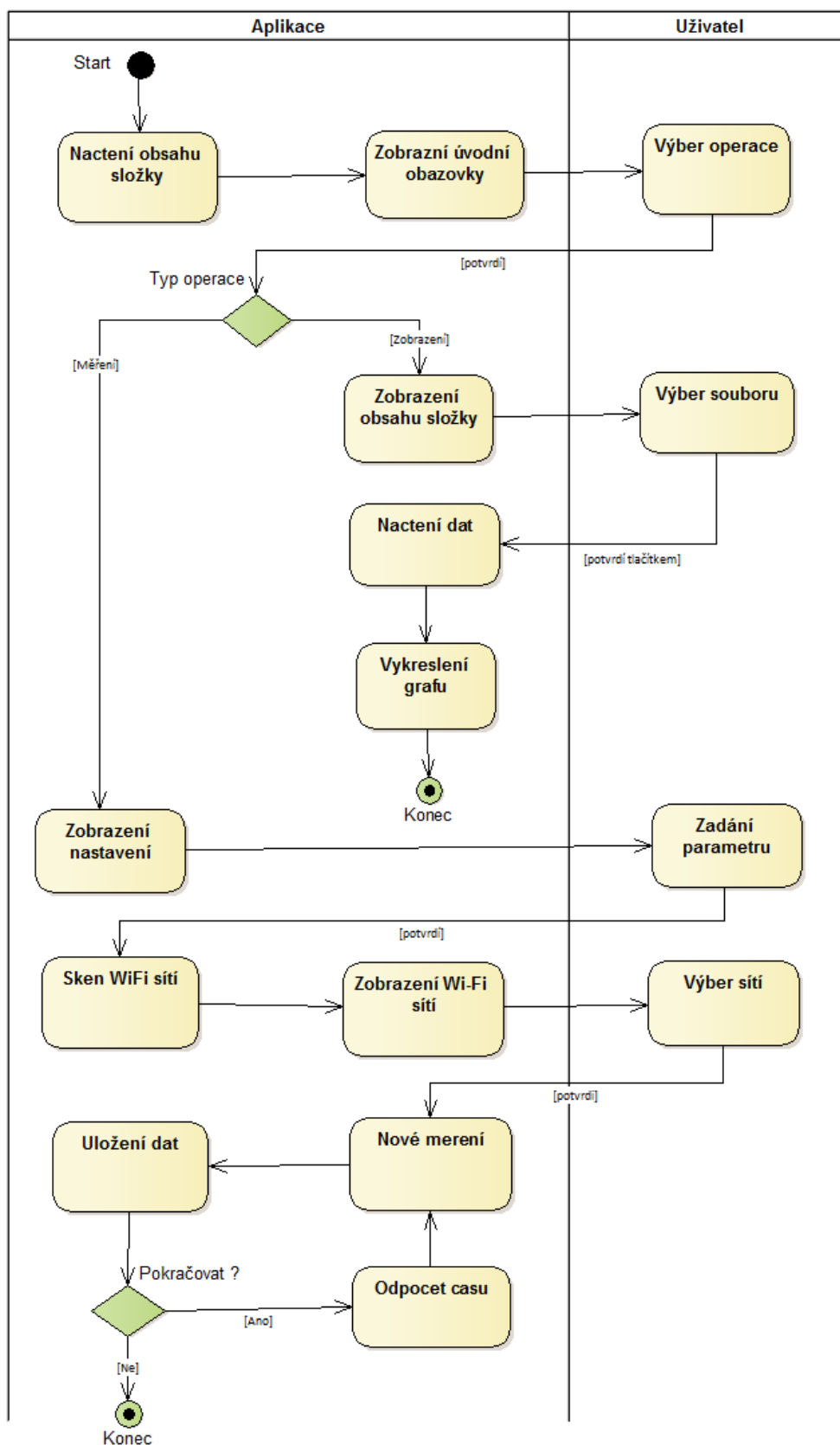
Zdroj: vlastní

4.2.5 Diagram aktivit

Diagram aktivit zachycuje průchod jednotlivými kroky, neboli workflow, od spuštění aplikace až po její skončení. Modeluje aktivity na vyšší úrovni abstrakce a nepopisuje technické podrobnosti, ani nezachycuje interakci mezi aplikací a operačním systémem. Dále je třeba vzít v potaz, že diagram zachycuje jen optimální

průchod jednotlivými kroky. Nepopisuje například situaci, kdy během jedné z aktivit nastane neočekávaná výjimka, a operační systém je přinucen celou aplikací restartovat nebo terminovat. Diagram aktivit na obrázku 18 modeluje průchod aktivitami současně pro případ užití měření propustnosti i zobrazení propustnosti.

Obrázek 19: Diagram aktivit

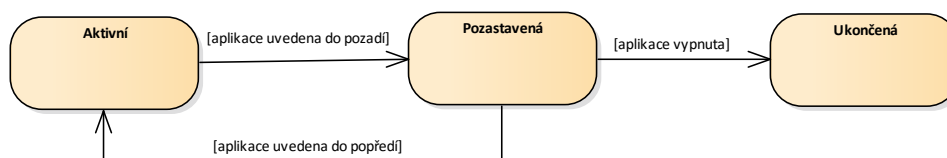


Zdroj: vlastní

4.2.6 Stavový diagram

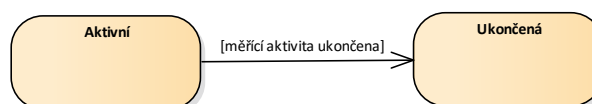
Operační systém Android spravuje stav jednotlivých aktivit sám. Například při překrytí spuštěné aplikace jinou aplikací, se původní aplikace dostane do pozadí a systém ji uvede do stavu „pauza“, a po delší době nečinnosti do stavu „stop“. Toto chování se však dá obejít nebo vynutit jeho změna. Následující stavový diagram na obrázku 19 modeluje, jakých stavů nabývá aktivita měření, a na obrázku jsou stavy servery, která běží asynchronně na pozadí a která provádí odpočet času a následné měření.

Obrázek 20: Stavový diagram měřící aktivity



Zdroj: vlastní

Obrázek 21: Stavový diagram měřící servery



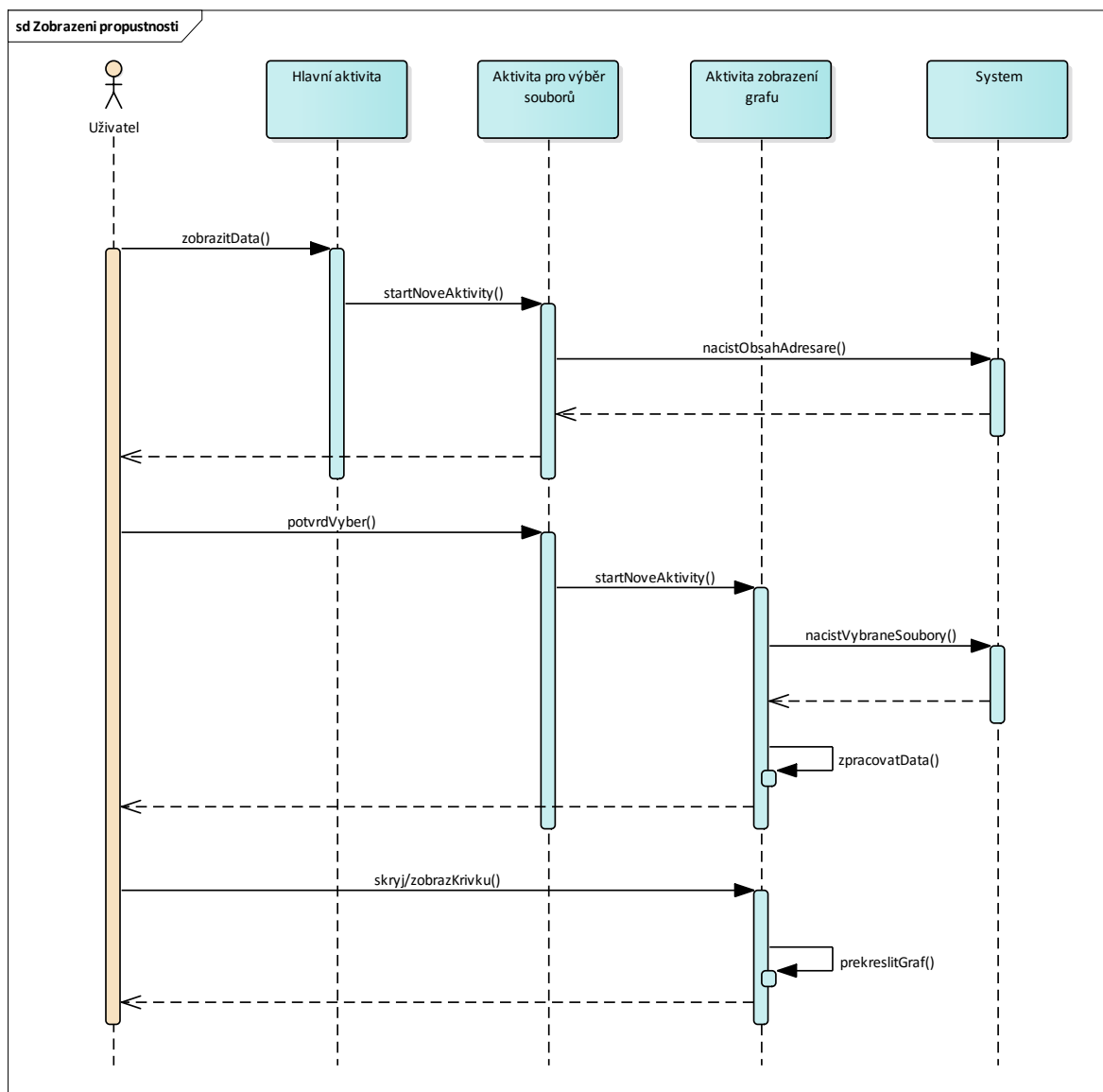
Zdroj: vlastní

4.2.7 Sekvenční diagram

Každá aplikace vyvíjená pro operační systém Android je složena z posloupnosti několika aktivit (Activity). Jednotlivé aktivity vykonávají určitou část logiky v konkrétním kroku užívání aplikace. Nastartování nové aktivity může být iniciováno například stiskem tlačítka. Každá z aktivit má definované vlastní zobrazení (View) které je načteno po startu aktivity. Sekvenční diagramy na

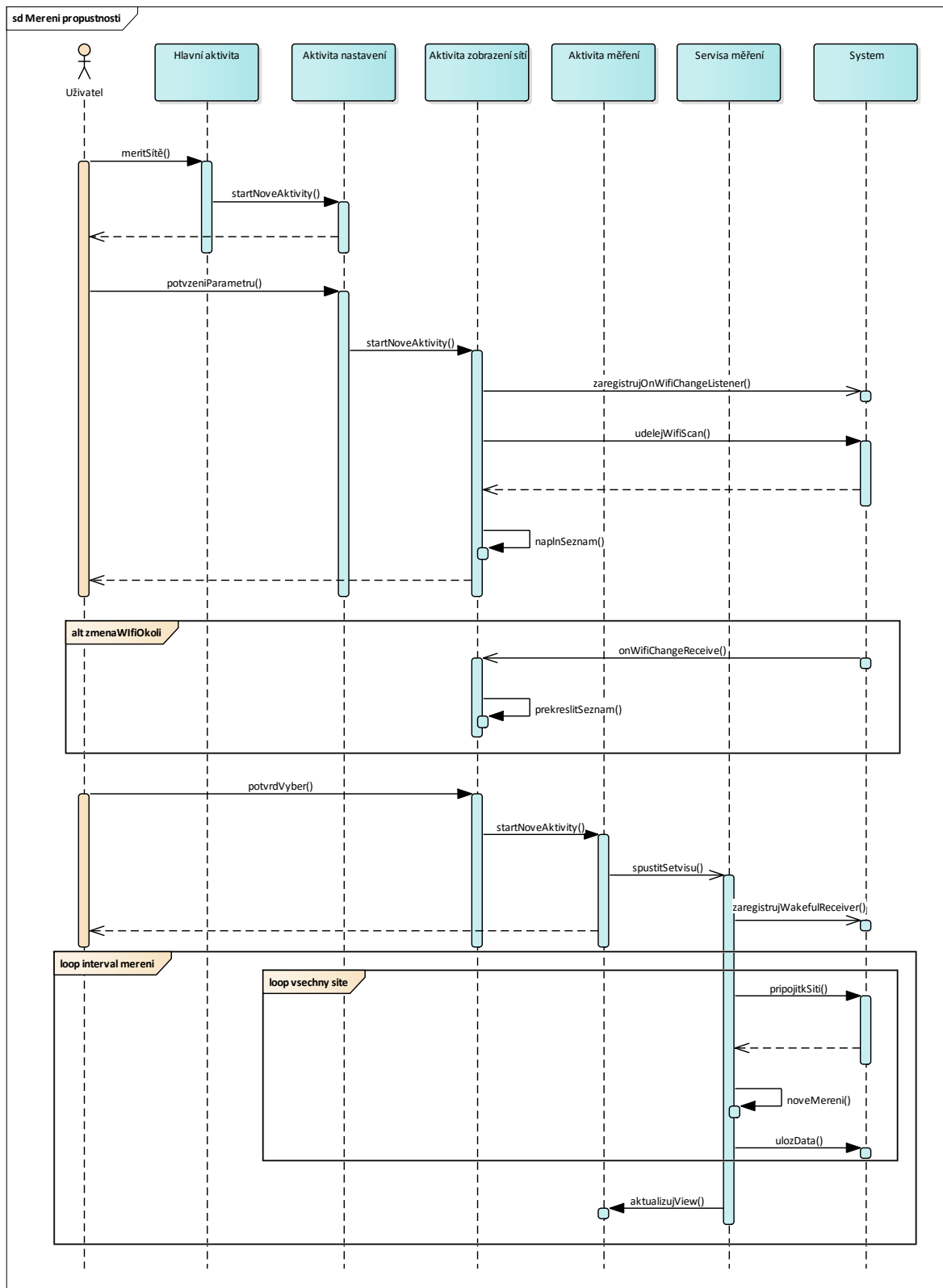
následujících dvou obrázcích znázornují jednotlivé aktivity budoucí aplikace, a jejich vzájemnou interakci v čase. Na obrázku 19 je modelována sekvence kroků pro případ užití kdy chce uživatel načíst a zobrazit uložená data, a na obrázku 20 lze vidět interakci jednotlivých aktivit a systému v případě užití, kde uživatel chce spustit nové měření.

Obrázek 22: Sekvenční diagram případu užití ‚zobrazení propustnosti‘



Zdroj: vlastní

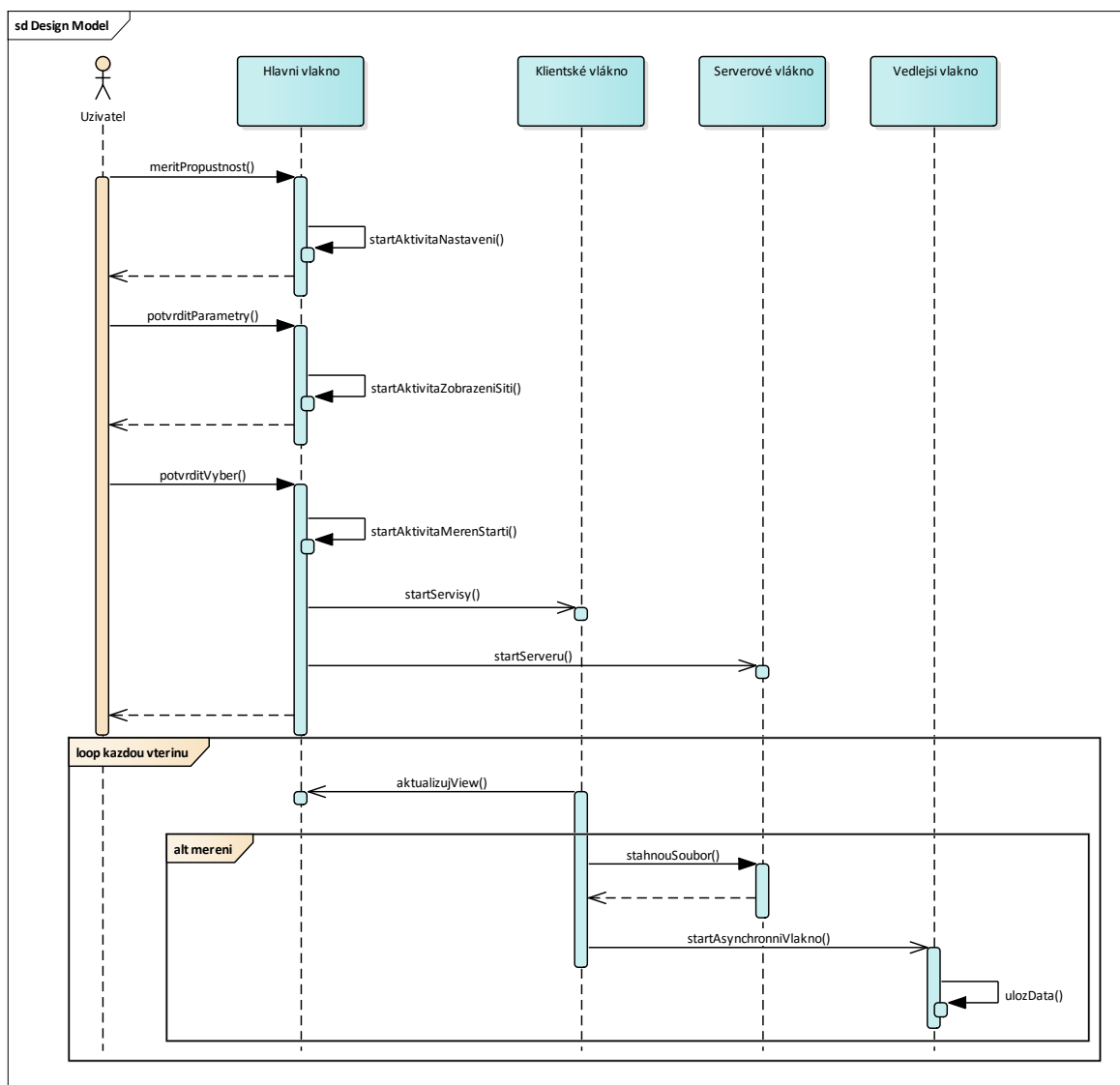
Obrázek 23: Sekvenční diagram případu užití „měření propustnosti“



Zdroj: vlastní

Jelikož je předpoklad, že aplikace bude běžet ve více vláknech, následující sekvenční diagram modeluje spouštění jednotlivých vláken a interakci mezi nimi. Model je pro případ užití měření propustnosti:

Obrázek 24: Sekvenční diagram aplikačních vláken



Zdroj: vlastní

4.3 Implementace

V této kapitole jsou slovně popsány zajímavé nebo klíčové části implementace navržené aplikace. Slovní popis je doplněn o reálné ukázky z prostředí Android Studio nebo přímo o grafický výstup prototypu aplikace.

4.3.1 Obecný přístup

Pro vývoj a správnou funkčnost Android aplikace je zapotřebí mít nadefinovaný manifest soubor, který popisuje jednotlivé aktivity a serვისy které tvoří aplikaci. V manifestu se také deklarují potřebná oprávnění, která jsou nezbytná pro fungování aplikace. Na následujícím obrázku je ukázka manifest.xml souboru:

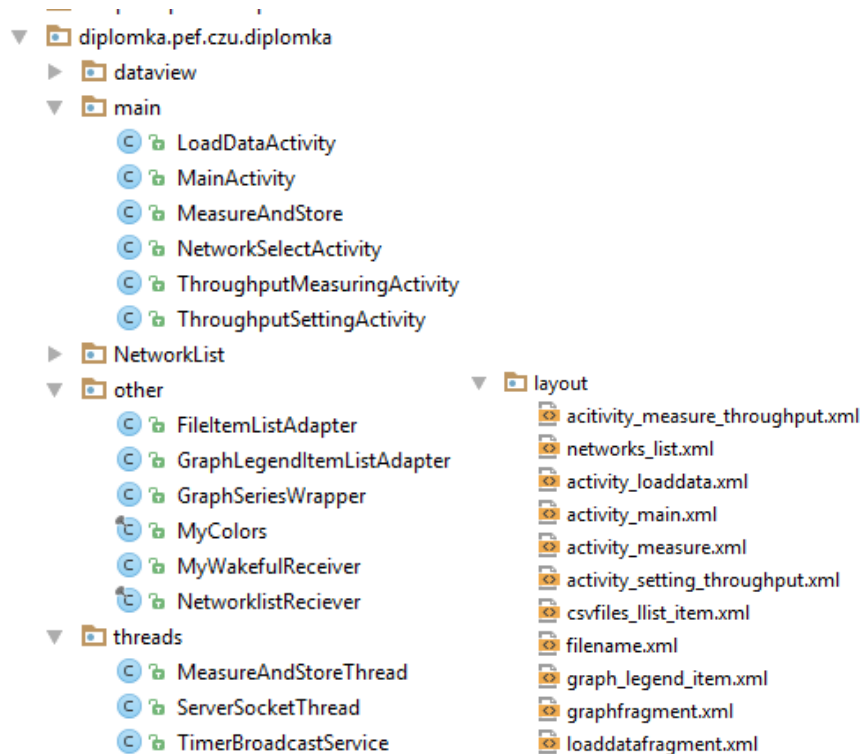
Obrázek 25: Ukázka manifest souboru

```
<!-- aktivita ThroughputMeasureActivity -->
<activity
    android:name="diplomka.pef.czu.diplomka.main.ThroughputMeasuringActivity"
    android:configChanges="orientation|screenSize"
    android:label="diplomka"
    android:parentActivityName="diplomka.pef.czu.diplomka.main.ThroughputSettingActivity" >
    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value="diplomka.pef.czu.diplomka.main.ThroughputSettingActivity" />
</activity>
<service android:name="diplomka.pef.czu.diplomka.threads.TimerBroadcastService" />
<receiver android:name="diplomka.pef.czu.diplomka.other.MyWakefulReceiver"/>
```

Zdroj: vlastní

Programovací jazyk Java používány při vývoji aplikací na platformě Android je objektový, a jednotlivé objekty a aktivity aplikace jdou modelovány pomocí tříd. Každá aktivita má vlastní view, a každé view je definováno pomocí samostatného layout xml souboru. Následující obrázek 25 nabízí ukázku členění tříd do balíků a definici layout souborů.

Obrázek 26: Strukturování souboru v nástroji Android Studio



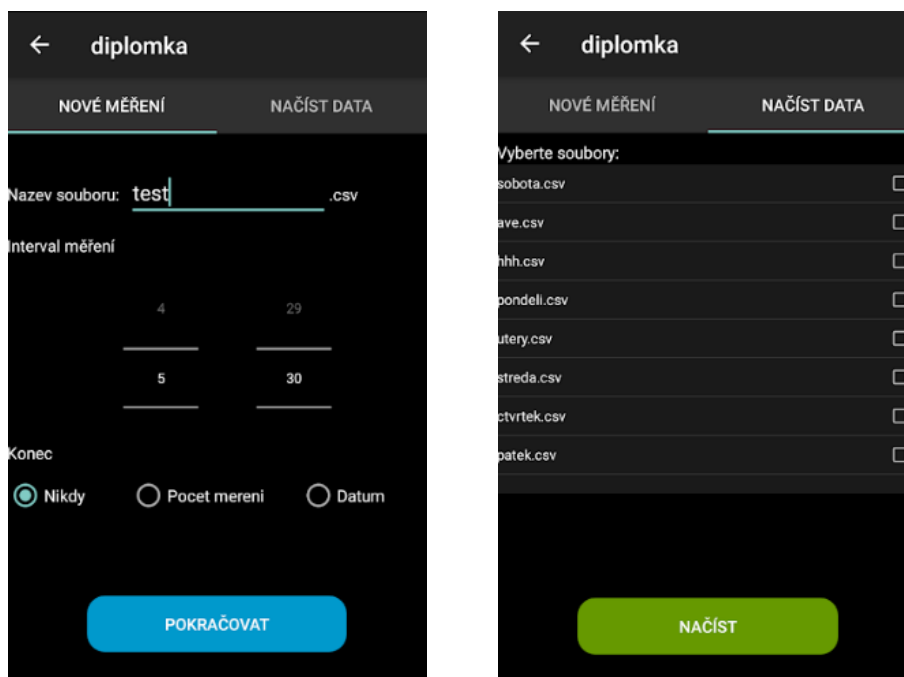
Zdroj: vlastní

4.3.2 Postup implementace

Fragmenty

Fragmenty umožňují definovat v rámci jedné aktivity více view. Toho přístupu se využilo ve vyvíjené aplikaci k tomu, aby uživatel mohl rychle přepínat mezi režimy, ve kterých chce aplikaci provozovat. Konkrétně mezi případem užití měření propustnosti a zobrazení propustnosti. Pro použití fragmentů bylo implementováno rozhraní `FragmentManagerAdapter` do kterého byly vloženy dvě view. Pro přepínání mezi jednotlivými view byla použita komponenta `TabLayout` reagující na uživatelův dotek displeje.

Obrázek 27: Ukázka fragmentů zobrazení



Zdroj: vlastní

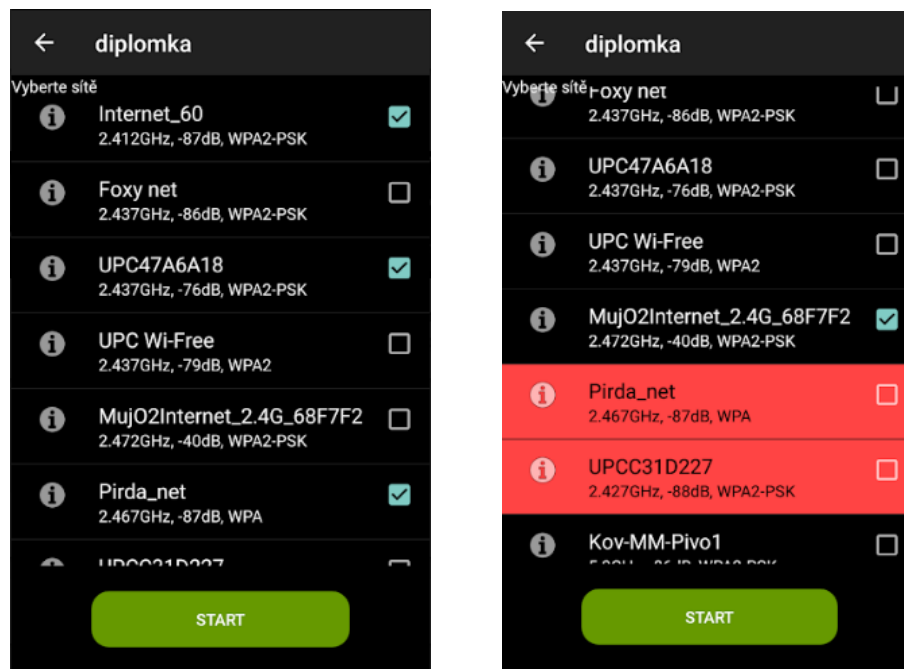
ListView

Pro zobrazení seznamu prvků bylo implementováno vlastní ListView a s tím související rozhraní ArrayAdapter. Účelem bylo, aby každý prvek seznam obsahoval nejen text, ale také zaškrťovací pole reagující na uživatelský dotek obrazovky, a umožňující výběr několika prvků ze seznamu. ListView bylo využito při zobrazení Wi-Fi sítí, csv souborů, a legendy grafu. Použití ListView je možné vidět na obrázku 26 a 27.

WifiChangeListener

Po zaregistrování WifiChangeListeneru systém informuje aplikaci pokud Wi-Fi čip detekuje jakoukoliv změnu v okolí. Listener je použit při zobrazení seznamu sítí v okolí. Po přijetí události od listeneru aplikace porovná nové hodnoty se stávajícími, a pokud z dosahu zmizí některá ze sítí, v seznamu se tato síť zvýrazní a není možné ji označit pro následné měření. Ukázkou lze vidět obrázku 27.

Obrázek 28: Ukázka ListView pro výběr sítí

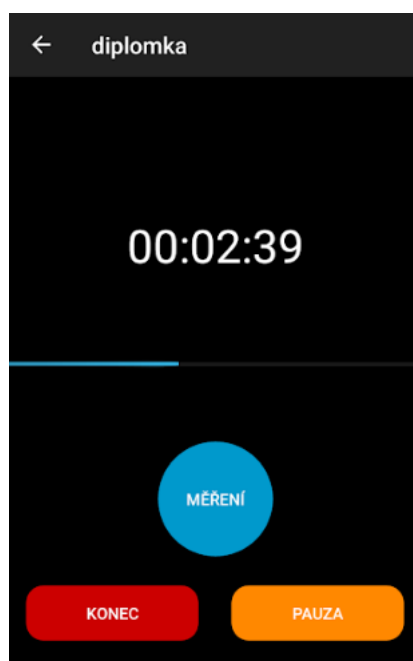


Zdroj: vlastní

Service

Aby měření mohlo probíhat i v případě že aplikace není momentálně na popředí nebo je zařízení neaktivní, bylo zapotřebí definovat takzvanou servisu (Service). Na rozdíl od standartních aktivit, servisy běží ve vlákně na pozadí i v případě že aktivita je v pozastaveném stavu. V našem případě byla servisa použita v měřící aktivitě a vykonává funkčnost čítače času a samotného měření propustnosti. Čítač času odpočítává čas mezi jednotlivými měřeními, a každou vteřinu asynchronně notifikuje rodičovskou aktivitu a ta aktualizuje své view se stopkami. Po dosažení času 0 provede měření a celý cyklus se opakuje. To vše se provede i v případě že zařízení je neaktivní a aktivity aplikací jsou uvedeny do pozastaveného stavu. Na obrázku je ukázka odpočtu času, který na pozadí řídí servisa.

Obrázek 29: Odpočet mezi měřeními



Zdroj: vlastní

Měření propustnosti

Implementace samotného měření byla vyřešena tak, že servisa po uběhnutí časového intervalu provede připojení k požadované síti. Následně nastartuje vlákno klienta, které se pomocí soketu (Socket) připojí k serverovému vláknu. Poté se mezi vlákny provede přenos souboru. Klient spočítá délku času přenosu od zahájení a po jeho dokončení a na základě velikosti souboru spočte přenosovou rychlost. I když je měření prováděno na jednom fyzickém zařízení, při přenosu souboru mezi vlákny, by každý přenášený paket měl být ze zařízení vyslán přes bezdrátový přístupový bod do směrovače, a přes bezdrátový bod opět zpět do zařízení.

Každé měření je provedeno hned pětkrát po sobě. Následně jsou naměřené hodnoty zprůměrovány a až poté uloženy. Celý cyklus se opakuje pro všechny sítě, které byly uživatelem vybrané k měření.

WakefulBroadcastReceiver

I když servisní vlákno běží na pozadí nepřetržitě, systém Android po delší nečinnosti zařízení vypíná z důvodu úspory energie veškeré vedlejší funkce. A to včetně Wi-Fi adaptéru. Z tohoto důvodu musel být v měřicí aktivitě zaregistrovaný WakefulBroadcastReceiver, který asynchronně na pozadí každou vteřinu posílá do systému budící příznak, a tím udržuje celý systém a jeho prvky ve stále aktivním stavu. A to i například pro dlouhodobě vypnutém a neaktivním displeji.

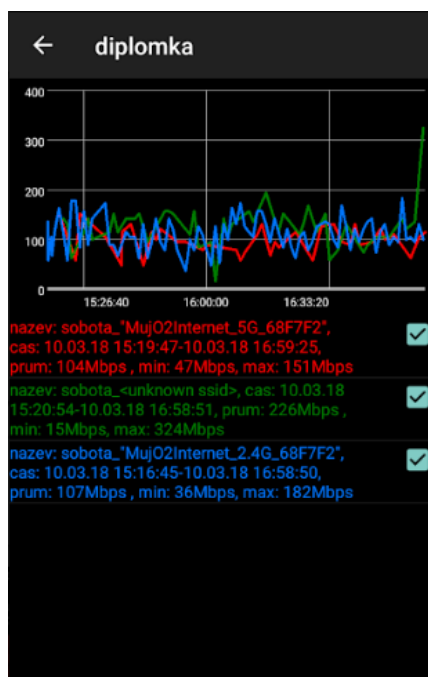
openCSV

Pro práci s CSV soubory byla použita open-source knihovna openCSV. Knihovna pracuje především s řetězcovými poli, které reprezentují řádky souboru. Prvky pole jsou pak jednotlivé položky v řádku. Zápis / čtení souboru včetně parsování obsahu a oddělovacích znaků obstarává knihovna.

GraphView

Pro vykreslení grafu byla použita open-source knihovna GraphView. Její funkcionality musela být drobně upravena. Jednalo se hlavně o požadavek na to, aby osa X vhodně pracovala s časem. Také bylo zapotřebí vyřešit posun křivek grafu u sítí, které byly naměřeny v různé dny. To znamená, že na ose není zohledněn časový rozdíl v rámci dnů. Pod graf byla přidána legenda, která kliknutím umožňuje zobrazení nebo skrytí jednotlivých křivek sítí. Opět zde byla použita vlastní implementace ListView. Legenda obsahuje doplňující informace ke konkrétní křivce grafu, jako jsou: název sítě, začátek a konec měření, průměrnou, minimální a maximální naměřenou hodnotu. Ukázka celé aktivity s grafem je zobrazena na obrázku 30.

Obrázek 30: Zobrazení propustnosti



Zdroj: vlastní

4.4 Testování

Cílem testování je ověření primárních požadavků a funkčností aplikace. Testy by měly především ověřit, že aplikace je schopna dlouhodobě měřit a ukládat propustnost vybraných sítí. Následně by mělo být otestováno, že aplikace korektně interpretuje naměřená data v podobě lineárního grafu.

4.4.1 Testovací scénář měření propustnosti

Tabulka 5: Testovací scénář měření propustnosti

Krok	Akce	Očekávaný výstup
1	Spust' aplikaci	Úvodní obrazovka

2	Vyber fragment NOVÉ MĚŘENÍ	Obrazovka nastavení
3	Zadej název souboru	-
4	Interval měření nastav na 10:00 minut	-
5	Stiskni tlačítko Pokračovat	Seznam dostupných sítí
6	Vyber síť k měření	Označení sítě
7	Stiskni tlačítko START	Odpočet času mezi měřeními
8	Nech aplikaci běžet několik hodin	-
9	Ukonči aplikaci	Soubor CSV s validními daty

Zdroj: vlastní

4.4.2 Testovací scénář zobrazení propustnosti

Tabulka 6: Testovací scénář zobrazení propustnosti

Krok	Akce	Očekávaný výstup
1	Spust' aplikaci	Úvodní obrazovka
2	Vyber fragment NAČÍST DATA	Seznam souborů ve složce
3	Vyber soubory	Označení souborů
4	Klikni na tlačítko načíst	Zobrazení grafu
5	Klikni na položku legendy	Zobrazení / skrytí sítě z grafu
6	Udělej gesto dvěma prsty	Změna měřítka grafu (zoom)

Zdroj: vlastní

5 Výsledky

Na základě návrhu byl naimplementován funkční prototyp aplikace. Pro ověření funkčnosti bylo následně v souladu s testovacími scénáři provedeno testování aplikace v reálných podmínkách. Testování probíhalo v průběhu několika vybraných dnů v týdnu (pondělí až pátek), a byla měřena vždy stejná Wi-Fi síť. Měření sítě v každém dni trvalo vždy několik hodin, a data byla ukládána do separátního souboru. Po celou dobu měření bylo zařízení ve všech testovacích dnech umístěno vždy na stejném místě. Pro měření byl použit mobilní telefon Sony Xperia Z3 Compact.

Výstupem každého měření v každém dni byl CSV soubor obsahující naměřená hodnoty. Po otevření souboru v libovolném tabulkovém procesoru můžeme zkontrolovat jeho obsah. Ukázka obsahu jednoho ze souborů je na obrázku 31:

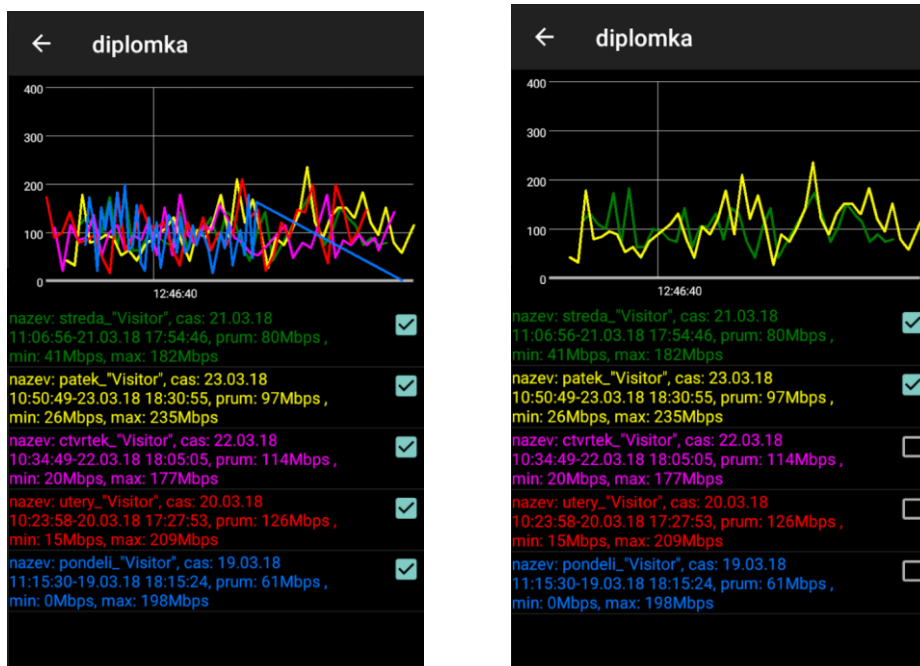
Obrázek 31: Ukázka obsahu csv souboru

1521633620510	visitor	70	13	5180	2003151040	200	38:0e:4d:d7:3d:c21
1521634235994	"Visitor"	73	7	5180	2003151040	200	38:0e:4d:d7:3d:c22
1521634857168	"Visitor"	141	3	5180	2003151040	200	38:0e:4d:d7:3d:c23
1521635463999	"Visitor"	62	2	5180	2003151040	200	38:0e:4d:d7:3d:c24
1521636099318	"Visitor"	104	9	5180	2003151040	200	38:0e:4d:d7:3d:c25
1521636706851	"Visitor"	104	26	5180	2003151040	200	38:0e:4d:d7:3d:c26
1521637329657	"Visitor"	130	1	5180	2003151040	200	38:0e:4d:d7:3d:c27
1521637936462	"Visitor"	78	12	5180	2003151040	200	38:0e:4d:d7:3d:c28

Zdroj: vlastní

Po provedení měření ve zvolených dnech byly všechny soubory pomocí aplikace načteny, a výstupem byl graf obsahující pět křivek – vždy jednu ke každé síti z každého načteného souboru. Kliknutím na vybrané síť v seznamu pod grafem pak můžeme zobrazit jen konkrétní křivky grafu. Například středu a pátek. Ukázku je možné vidět na obrázku 32.

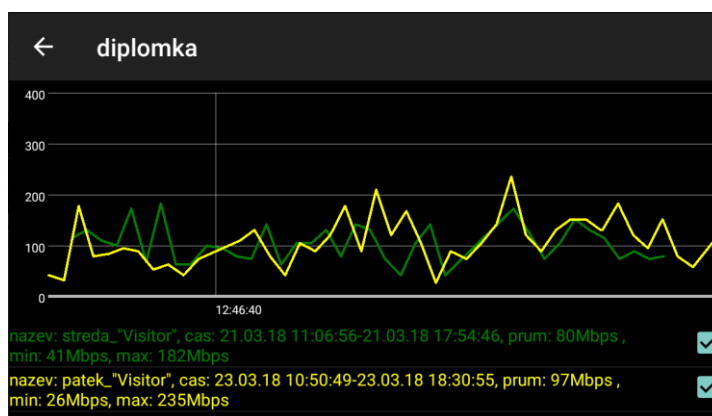
Obrázek 32: Test aplikace – zobrazení grafu



Zdroj: vlastní

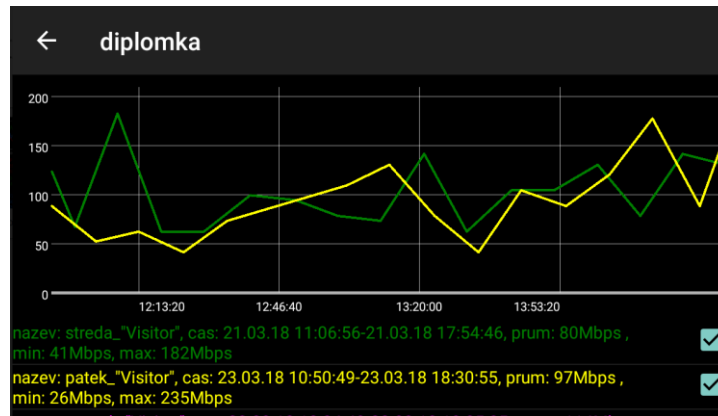
Otočením zařízení lze graf zobrazit na šířku, a gestem dvou prstů lze změnit měřítko osy X, a zaměřit se tak na podrobnější časový úsek. Ukázky jsou uvedeny na obrázcích 32 a 33.

Obrázek 33: Test aplikace – zobrazení grafu na šířku



Zdroj: vlastní

Obrázek 34: Test aplikace – zobrazení grafu s přiblížením



Zdroj: vlastní

V první fázi tedy nejdříve probíhalo pomocí aplikace měření a sběr dat. Výstupem byly csv soubory obsahující naměřené hodnoty. Poté byla tato data aplikací znovu načtena, a výstupem byl graf o několika křivkách, Každá křivka odpovídá jedné síti v každém z načtených souborů. Ukázka grafu je na obrázku 31. Pro vhodnou přehlednost byly některé křivky graf skryty a graf byl otočením mobilního telefonu zobrazen na šířku a pomocí gesta dvou prstů bylo zaměřeno na kratší časový úsek křivky. Z obrázků 33 a 34 lze přehledně vidět srovnání hodnot propustnosti stejné sítě v průběhu dvou dnů.

V následující tabulce je uveden přehled požadavků na vyvíjenou aplikaci, a na základě otestování aplikace je u každého požadavku uvedeno, zda byl splněn, či nikoliv:

Tabulka 7: Naplnění požadavků

Požadavek	Zhodnocení
Nativní mobilní aplikace	splněno
Přehledné uživatelské rozhraní	neověřeno
Intuitivní ovládání	neověřeno
Stabilita chování	splněno
Přenositelnost dat	splněno
Minimalistická	neověřeno

Měření propustnosti Wi-Fi sítě	splněno
Zobrazení propustnosti	splněno
Měření jedním zařízením	splněno
Automatické měření	splněno
Ruční měření	splněno
Přenosnost naměřených dat	splněno
Perzistence dat	splněno

Zdroj: vlastní

Lze konstatovat, že většina požadavků z definovaných ve fázi návrhu byla splněna a naplnila se tak vize a cíl vyvíjené aplikace. U několika nefunkčních požadavků nelze stanovisko s určitostí rozhodnout. Například přehlednost a intuitivnost nelze objektivně posoudit a výsledky by se daly vyhodnotit až po pilotním provozu aplikace reálnými uživateli. Stejně tak určení zda je aplikace opravdu minimalistická by vyžadovalo provedení zátěžových testů a monitorování vytížení jednotlivých částí operačního systému.

6 Závěr

Hlavním cílem této diplomové práce byl vývoj mobilní aplikace, která bude měřit a vizualizovat propustnost bezdrátových. V souladu s dílčími cíli práce byla nejdříve sestavena literární rešerše zaměřující se na problematiku vývoje mobilních aplikací a Wi-Fi sítí. Literární rešerše nejdříve definuje a popisuje charakteristické vlastnosti mobilních aplikací, jejich členění, a odlišnost od desktopových aplikací. Následně byla popsána metodika a doporučené postupy pro úspěšný vývoj funkčního prototypu mobilní aplikace. Pro přehled jsou v ní také uvedeny a porovnány tři nejčastěji zastoupené mobilní platformy. Druhá část zpracované literární rešerše je zaměřena na bezdrátové sítě. Nejdříve je popsáno dělení bezdrátových sítí a jejich charakteristiky, a následně jsou blíže specifikovány a popsány vlastnosti a veličiny Wi-Fi sítí.

Teoretická část práce byla zaměřena na návrh, implementaci a následné otestování aplikace. Na základě vize a provedené analýzy konkurenčních aplikací byly stanoveny požadavky na budoucí aplikaci. Za pomoci grafického jazyka UML byly vymodelovány diagramy popisující strukturu a chování vyvíjené aplikace. Kompletní zdrojový kód implementace není v textu práce obsažen, ale jsou zde slovně popsány její dílčí kroky, které vedli k řešení úlohy této práce. Slovní popis byl doplněn o grafické ukázky jednotlivých částí funkčního prototypu aplikace.

Pro ověření funkčnosti aplikace a splnění definovaných funkčních požadavků byla aplikace podrobená testům v reálném prostředí. Testování probíhalo v rámci několika dnů ve stejném prostředí a na stejné síti. Výstupem testování byl graf s několika křivkami, který umožňuje srovnání průběhu propustnosti jedné sítě v průběhu několika různých dní.

Výstupy testu byly uvedeny v závěru a práce, a na jejich základě byla provedena syntéza výsledků a výstupů práce.

7 Seznam použitých zdrojů

BELLALTA, Boris. Next generation IEEE 802.11 Wireless Local Area Networks: Current status, future directions and open challenges. [online] 2016 [cit. 10.3.2018]. Dostupné z: <https://www-sciencedirect-com.infozdroje.czu.cz/science/article/pii/S0140366415003874#bib0052>

CAVALHEIRO, Gerson. PILLA, Maurício Camargo, C.A.S. Aproximating static list schedules in dynamic multithreaded applications. [online] 2014 [cit. 10.3.2018]. ISSN: 1573-7543 .Dostupné z: <https://doi-org.infozdroje.czu.cz/10.1007/s10586-013-0322-3>

CURCIO, Karina, NAVARRO, Tiagro, MALUCELLI, Andreia. Requirements engineering: A systematic mapping study in agile software development. [online] 2018 [cit. 10.3.2018]. Dostupné z: <https://www-sciencedirectcom.infozdroje.czu.cz/science/article/pii/S0164121218300141>

DOBKIN, Daniel M.. RF Engineering for Wireless Networks : Hardware, Antennas, and Propagation. [online] 2005 [cit. 10.3.2018]. Dostupné z: <https://ebookcentral-proquest-com.infozdroje.czu.cz/lib/czup/detail.action?docID=286738>

FLOOD, Harrison R., DUCE, D. Usability of mobile applications: literature review and rationale for a new usability model [online]. 2013 [cit. 10.3.2018]. ISSN 2194-0827. Dostupné z: <https://link.springer.com/article/10.1186/2194-0827-1-1>

FLORA, Harleen, K., WANG, Xiaofeng, CHANDE, Swati V. An Investigation on the Characteristics of Mobile Applications: A Survey Study [online]. 2014 [cit. 10.3.2018]. Dostupné z: https://www.researchgate.net/profile/Harleen_Flora6/publication/266743359_An_Investigation_on_the_Characteristics_of_Mobile_Applications_A_Survey_Study/

FLORA, Harleen, K., WANG, Xiaofeng, CHANDE, Swati V. An Investigation into Mobile Application Development Processes: Challenges and Best Practices [online]. 2014

[cit. 10.3.2018]. Dostupné z: https://www.researchgate.net/publication/266743674_An_Investigation_into_Mobile_Application_Development_Processes_Challenges_and_Best_Practices

GAO, Steven. LUO, Qi, ZHU, Fuguo. Circularly Polarized Antennas. [online] 2013 [cit. 10.3.2018]. ISBN: 9781118790502. Dostupné z: <https://ebookcentral-proquest-com.infozdroje.czu.cz/lib/czup/detail.action?docID=1527856>.

GARTNER. Gartner Says Emerging Markets Drove Worldwide Smartphone Sales to 19 Percent Growth in First Quarter of 2015. [online] 2015 [cit. 10.3.2018]. Dostupné z: <http://www.gartner.com/newsroom/id/3061917>

GAST, Matthew. 802.11 wireless networks: The definitive guide. 2006 [cit. 10.3.2018].

HOLÍK, Filip. Metodika měření propustnosti bezdrátové sítě. [online] 2014 [cit. 10.3.2018]. Dostupné z: <http://dspace.upce.cz/handle/10195/55904>

HOLT, Alan, HUANG, Chi-Yu . 802.11 Wireless Networks. [online] 2010 [cit. 10.3.2018]. ISBN: 978-1-84996-275-9. Dostupné z: <https://doi-org.infozdroje.czu.cz/10.1007/978-1-84996-275-9>

KANIKA, Sharma. A Study of Wireless Networks: WLANs, WPANs, WMANs, and WWANs with Comparison. [online] 2015 [cit. 10.3.2018]. ISSN: 0975-9646. Dostupné z: <https://pdfs.semanticscholar.org/ff3e/8a75932416553f16adf113245c1842a0f09b.pdf>

LANO, Kevin. WILEY, John. UML 2 Semantics and Applications. [online] 2014 [cit. 10.3.2018]. Dostupné z: <https://ebookcentral-proquest-com.infozdroje.czu.cz/lib/czup/detail.action?docID=711732> .

LO, Cheng-Chung E. An Investigation of the Impact of Signal Strength on Wi-Fi Link Throughput through Propagation Measurement. [online] 2007 [cit. 10.3.2018]. Dostupné z: <http://aut.researchgateway.ac.nz/bitstream/handle/10292/698/LoE.pdf?sequence=1&isAllowed=y>

MCWHERTER, Jeff, GOWELL Scott. Professional Mobile Application Development, John Wiley & Sons, Incorporated, 2012 [cit. 10.3.2018].. Dostupné z: <https://ebookcentral-proquest-com.infozdroje.czu.cz/lib/czup/detail.action?docID=843643>.

MOUMANE, Idri K., ABRAN, A. Usability evaluation of mobile applications using ISO 9241 and ISO 25062 standards [online]. 2016 [cit. 10.3.2018]. ISSN: 2193-1801. Dostupné z: <https://link.springer.com/article/10.1186/s40064-016-2171-z>

PRASAD, Neeli. PRASAD, Annad. 802.11 WLANs and IP Networking : Security, QoS, and Mobility. [online] 2005 [cit. 10.3.2018]. Dostupné z: <https://ebookcentral-proquest-com.infozdroje.czu.cz/lib/czup/detail.action?docID=231634>.

SHUANG, Song, BIJU, Issac. Analysis of WiFi and WiMAX and Wireless Network Coexistence. [online] 2015 [cit. 10.3.2018]. ISSN 0974-9322. Dostupné z: <https://arxiv.org/abs/1412.0721>

VITHANI Tejas, KUMAR, Anand. Modeling the Mobile Application Development Lifecycle [online]. 2014 [cit. 10.3.2018]. ISSN: 2078-0966. Dostupné z: http://www.iaeng.org/publication/IMECS2014/IMECS2014_pp596-600.pdf

ZANDL, Patrick. Bezdrátové sítě WiFi Praktický průvodce. 2003 [cit. 10.3.2018]. ISBN 80-7226-632-2.

ZIKMUND, Jan. Srovnání přístupů multiplatformního vývoje mobilních aplikací [online]. 2015 [cit. 10.3.2018]. Dostupné z: https://theses.cz/id/cddonx/ISIS_49192_xzikj00.pdf?lang=en

Seznam obrázků

Obrázek 1: Kategorizace a procentuální zastoupení aplikací pro platformu Android.....	18
Obrázek 2: Fáze identifikace	24
Obrázek 3: Fáze designu.....	25
Obrázek 4: Vývojová fáze	26
Obrázek 5: Fáze prototypování.....	26
Obrázek 6: Fáze testování.....	27
Obrázek 7: Vývoj zastoupení mobilních platforem na trhu (2017).....	28
Obrázek 8: Přehled bezdrátových sítí v závislosti na mobilitě a přenosové rychlosti.....	33
Obrázek 9: Vizualizace signálu ze směrové antény.....	34
Obrázek 10: Vizualizace ze všesměrové antény.....	35
Obrázek 11: Modulace FHSS	39
Obrázek 12: Modulace DSSS	40
Obrázek 13: Ortogonální uspořádání kanálu OFDM.....	41
Obrázek 14: Výměna paketů při sdílení media.....	42
Obrázek 15: Skutečné využití celkové propustnosti Wi-Fi sítě.....	44
Obrázek 16: Ukázka aplikace TamoSoft Throughput	46
Obrázek 17: Ukázka aplikace Network Speed Tester	47
Obrázek 18: Diagram případů užití	51
Obrázek 19: Diagram aktivit.....	53
Obrázek 20: Stavový diagram měřící aktivity	54
Obrázek 21: Stavový diagram měřící služby	54
Obrázek 22: Sekvenční diagram případu užití ‚zobrazení propustnosti‘	55
Obrázek 23: Sekvenční diagram případu užití ‚měření propustnosti ‘	56
Obrázek 24: Sekvenční diagram aplikačních vláken	57
Obrázek 25: Ukázka manifest souboru	58
Obrázek 26: Strukturování souboru v nástroji Android Studio	59
Obrázek 27: Ukázka fragmentů zobrazení.....	60
Obrázek 28: Ukázka ListView pro výběr sítí	61
Obrázek 29: Odpočet mezi měřeními	62
Obrázek 30: Zobrazení propustnosti.....	64
Obrázek 31: Ukázka obsahu csv souboru	66
Obrázek 32: Test aplikace – zobrazení grafu.....	67
Obrázek 33: Test aplikace – zobrazení grafu na šířku.....	67
Obrázek 34: Test aplikace – zobrazení grafu s přiblížením.....	68

Seznam tabulek

Tabulka 1: Typy mobilních aplikací	15
Tabulka 2: Kategorie mobilních aplikací.....	17
Tabulka 3: Srovnání bezdrátových sítí	32
Tabulka 4: Přehled verzí standardu IEEE 802.11	36
Tabulka 5: Testovací scénář měření propustnosti.....	64
Tabulka 6: Testovací scénář zobrazení propustnosti	65
Tabulka 7: Naplnění požadavků	68