



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

DEPARTMENT OF COMPUTER SYSTEMS

**MONITOROVÁNÍ CELISTVOSTI SKUPINY POMOCÍ
BLUETOOTH**

GROUP COMPLETENESS MONITORING VIA BLUETOOTH

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

HYNEK HODULÍK

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. ZDENĚK VAŠÍČEK, Ph.D.

BRNO 2021

Zadání bakalářské práce



Student: **Hodulík Hynek**
Program: Informační technologie
Název: **Monitorování celistvosti skupiny pomocí Bluetooth
Group Completeness Monitoring via Bluetooth**
Kategorie: Počítačové sítě

Zadání:

1. Seznamte se s technologií Bluetooth a variantami vhodnými pro nízkoenergetickou komunikaci mezi více prvky, jako je např. Bluetooth Low Energy nebo Bluetooth Mesh.
2. Navrhněte systém pro monitorování celistvosti skupiny využitelný např. v oblasti cyklistiky, který v reálném čase provádí sledování dostupnosti členů skupiny a informuje v případě odtržení některého z členů.
3. Navržený systém implementujte formou aplikace pro zařízení s OS Android. Při návrhu komunikačního protokolu dbejte na minimalizaci energetické náročnosti řešení a bezpečnost řešení z pohledu ochrany osob proti sledování.
4. Vyhodnoťte a diskutujte vlastnosti navrženého řešení.

Literatura:

- Dle pokynů vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Splnění bodů 1 a 2 zadání.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Vašíček Zdeněk, doc. Ing., Ph.D.**

Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.

Datum zadání: 1. listopadu 2020

Datum odevzdání: 12. května 2021

Datum schválení: 30. října 2020

Abstrakt

Cílem této práce je vytvořit mobilní aplikaci pro operační systém Android, která bude sloužit k vytvoření a následnému sledování skupiny lidí při sportu jako je cyklistika nebo turistika. Aplikace má za úkol informovat o oddělení některého člena. Ke komunikaci je využita technologie Bluetooth Low Energy, pomocí níž zařízení vytvoří síť s mesh topologií. Aplikace je implementována v programovacím jazyce Java.

Abstract

The aim of this thesis is to create smartphone application for Android operating system. This application will serve to create and then monitor a group of people during an activity like cycling or hiking. The main purpose of the application is to inform others when a member separates from the rest of the group. Devices communicate through Bluetooth Low Energy and form a network with mesh topology. The application is implemented in Java programming language.

Klíčová slova

Android, Bluetooth, Bluetooth Low Energy, BLE, Java, Mesh

Keywords

Android, Bluetooth, Bluetooth Low Energy, BLE, Java, Mesh

Citace

HODULÍK, Hynek. *Monitorování celistvosti skupiny pomocí Bluetooth*. Brno, 2021. Bachelářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. Ing. Zdeněk Vašíček, Ph.D.

Monitorování celistvosti skupiny pomocí Bluetooth

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana doc. Ing. Zdeňka Vašíčka, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Hynek Hodulík
11. května 2021

Poděkování

Chtěl bych poděkovat vedoucímu práce panu Doc. Ing. Zdeňku Vašíčkovi, Ph.D. za odborné vedení, cenné připomínky a konzultace, které mi poskytl při vypracování této práce. Dále bych chtěl poděkovat své rodině za poskytnutou podporu.

Obsah

1	Úvod	2
2	Bluetooth	3
2.1	Princip komunikace	3
2.2	Verze	4
3	Bluetooth Low Energy	6
3.1	Porovnání s klasickým Bluetooth	6
3.2	Architektura	6
3.3	Princip komunikace	10
4	Bluetooth Mesh	13
4.1	Vznik	13
4.2	Princip komunikace	13
4.3	Architektura	14
4.4	Topologie	16
4.5	Neoficiální Bluetooth síť	17
5	Prostředí Android	19
5.1	Podpora Bluetooth Low Energy	19
5.2	Rozhraní pro práci s Bluetooth Low Energy	19
5.3	Oprávnění	21
5.4	Služby	21
6	Návrh řešení	23
6.1	Celkový princip	23
6.2	Způsob komunikace mezi zařízeními	24
6.3	Sledování členů skupiny	26
6.4	Bezpečnost	29
6.5	Grafické rozhraní aplikace	30
7	Implementace	31
7.1	Použité implementační nástroje	31
7.2	Struktura projektu	32
7.3	Vyhodnocení vlastností navrženého řešení na reálných zařízeních	35
8	Závěr	38
	Literatura	39

Kapitola 1

Úvod

Cyklistiku si, jako spoustu aktivit, nejvíce užijeme s přáteli. Ve skupinách se ale naráží na problémy s výkonností, někteří cyklisté jezdí rychleji, některým dojdou síly a jedou pomalu. Může se stát, že rychlejší část skupiny ujede ostatním. Tato bakalářská práce se zabývá vytvořením aplikace pro mobilní telefony s operačním systémem Android, která tomu zabrání. Při odjetí čela skupiny nebo problému někoho vzadu, ostatním zobrazí upozornění, že něco není v pořádku a je třeba zpomalit nebo zastavit.

Android jsem zvolil jako cílovou platformu, protože chytrý telefon je velmi rozšířen a spousta lidí ho používá jako cyklopočítač, takže se jedná o logické doplnění funkcionality. Aplikace ke komunikaci využívá Bluetooth, tím pádem nespotřebovává mobilní data a nekomunikuje skrze žádný vzdálený server. Informuje čistě při ztrátě signálu od některého člena skupiny.

Obsah této práce je rozdělen do kapitol podle logických celků. První tři kapitoly jsou zaměřeny na technologii Bluetooth. Konkrétně kapitola 2 se zabývá Bluetooth Classic, kapitola 3 Bluetooth variantou Low Energy, kladoucí důraz na nízkou spotřebu energie a poslední z těchto tří kapitol, kapitola 4 se zabývá Bluetooth Mesh - variantou Bluetooth sloužící ke komunikaci více zařízení. Následuje kapitola 5 zabývající se platformou Android. Kapitola 6 se zabývá návrhem aplikace od způsobu komunikace přes vyhodnocování ztráty po grafické rozhraní. Kapitola 7 se věnuje implementaci této aplikace a jejímu testování. Poslední kapitolou je závěr.

Kapitola 2

Bluetooth

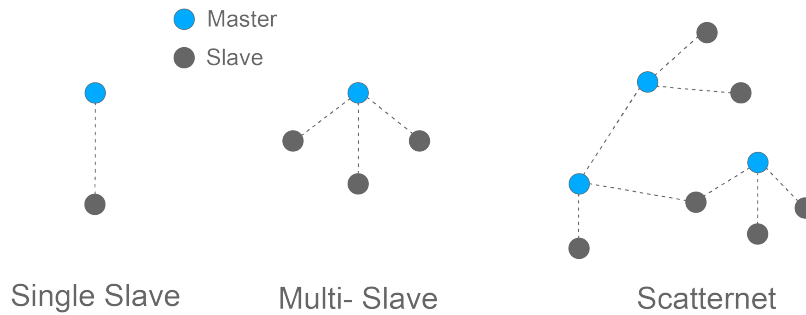
Bluetooth je standard bezdrátové komunikace, který je původně navržen pro komunikaci na krátké vzdálenosti. Slouží ke komunikaci mezi zařízeními jako jsou chytré telefony, sluchátka, počítače, klávesnice atd. Pro přenos signálu používá rádiové vlny na frekvenci 2.4 GHz. Motivací stojící za jeho vznikem bylo vytvoření levné technologie, která nahradí komunikaci počítače nebo telefonu s periferiemi pomocí drátu. O vývoj a specifikaci standardu se stará Bluetooth Special Interest Group (SIG). Tato skupina se skládá z firem jako je například Ericsson, který Bluetooth původně vyvinul, Nokia, Intel, Toshiba, Apple, Lenovo a asi 36 000 dalších. V průběhu existence Bluetooth došlo k vytvoření specifikací, které rozšiřují původní použití Bluetooth. Jedná se o Bluetooth Low Energy a Bluetooth Mesh, kterým se věnují následující kapitoly.

2.1 Princip komunikace

Bluetooth využívá nelicencované ISM pásmo mezi 2.402 GHz a 2.480 GHz. Stejně pásmo využívají i další technologie jako Wi-Fi, ZigBee nebo bezdrátové myši a klávesnice. Používané pásmo je pro potřeby Bluetooth dále rozděleno na 79 kanálů šířky 1Mhz. Aby byl minimalizován vliv rušení v tomto protěžovaném pásmu, je používána technika *frequency-hopping*. Zařízení tak pravidelně mění frekvenci, na které vysílá. Změnu může udělat až 1600 krát za sekundu [3].

Topologie

Bluetooth Classic, jak je tato technologie nyní nazývána, umožňuje point-to-point komunikaci. Probíhá tedy mezi dvěma zařízeními. Jedno z těchto zařízení je master a druhé slave. K jednomu master zařízení může být najednou připojeno až 7 aktivních slave zařízení. Taková síť se nazývá piconet. Jednotlivé piconety lze dále sdružovat, tímto spojením vznikne scatternet. Možné topologie jsou znázorněny na obrázku 2.1. K identifikaci zařízení Bluetooth slouží 48 bitové adresy. Ty jsou využívány zejména při párování. Po připojení master přidělí slave 3 bitovou adresu, která ho bude identifikovat v rámci piconetu [3].



Obrázek 2.1: Topologie sítí, které mohou vytvářet Bluetooth zařízení [18].

Synchronizace

Důležitým aspektem Bluetooth je časování. Je třeba spolehlivě určit časový průběh jednotlivých operací. Jejich synchronizace je důležitá, aby bylo možné spolehlivě vysílat a přijímat data posílaná mezi master a slave. Každý slave tedy synchronizuje své hodiny s hodinami zařízení master.

Komunikační kanál je rozdělen na $625\mu s$ intervaly. V těchto časových intervalech zařízení střídavě vysílají. Master využívá sudé intervaly, slave liché. Jeden paket může být rozdělen do pěti těchto tzv. slotů [3].

Profily

Bluetooth kromě definice komunikačních protokolů definuje také profily. Ty slouží ke zjednodušení práce s Bluetooth, definují totiž druh dat, která jsou přenášena. Profil je určen použitím zařízení, existují například profily pro přenos hudby nebo souborů. Aby spolu mohla dvě zařízení komunikovat, musejí podporovat stejné profily [3].

2.2 Verze

Bluetooth, po dobu své více než dvacetileté existence, prošel značným vývojem. Jeho, v současné době, 5 hlavních verzí přineslo řadu vylepšení co se týče zvyšování rychlosti a dosahu, ale i vytváření dalších řešení reagujících na měnící se poptávku na trhu (Low Energy, Mesh) [5].

Bluetooth 1.0

Verze 1.0 byla první verze Bluetooth. Ve své době to byla přelomová technologie, ale jako každá prvotní verze trpěla svými nedostatky. Hlavním problémem byla kompatibilita mezi zařízeními různých výrobců. Tato verze také používala pouze fyzickou MAC adresu zařízení určenou výrobcem, což znemožňovalo anonymní komunikaci.

Tyto chyby byly postupně odstraněny ve verzích 1.1 a 1.2.

Bluetooth 2.0 + EDR

Hlavní novinkou této verze bylo EDR - Enhanced Data Rate umožňující zvýšení rychlosti přenosu dat. Maximální rychlost přenosu tím byla zvýšena na 2,1 Mb/s.

Bluetooth 3.0 + HS

Tato verze umožňuje komunikaci až rychlostí 24 Mb/s. Té je dosaženo použitím Bluetooth pouze k navázání spojení a samotný přenos dat je realizován pomocí Wi-Fi.

Bluetooth 4.0

Bluetooth 4.0 je důležitou verzí, protože specifikace 4.0 definuje Bluetooth Low Energy (BLE), který existuje spolu s původní verzí nově označovanou jako Bluetooth Classic. Hlavním přínosem verze Low Energy je nízká spotřeba umožňující použití v zařízeních napájených z malých baterií. BLE se podrobněji věnuje následující kapitola [3](#).

Bluetooth 5.0

Verze 5.0 se opět soustředí na zvýšení přenosové rychlosti a zvětšení dosahu komunikace. A to zejména pro použití v Internet of Things. Zajímavá je i verze 5.1, která se soustředí na použití Bluetooth pro navigaci. Tato verze dokáže určit polohu zařízení na základě úhlu, ve kterém od něj přijímá signál.

Kapitola 3

Bluetooth Low Energy

Bluetooth Low Energy (BLE) je verze Bluetooth vyvinutá společností Nokia. Tehdy se ještě jmenoval Wibree a byl integrován do specifikace Bluetooth 4.0 pod správou organizace Bluetooth Special Interest Group (Bluetooth SIG). BLE slouží k nízkoenergetické komunikaci mezi zařízeními, u kterých se nečeká vysoká rychlost přenosu. Nízké spotřeby je dosaženo hlavně díky tomu, že malé objemy dat, které se přes BLE posílají, trvá poslat jen krátký časový okamžik (řádově milisekundy), ve zbývajícím čase je zařízení v úsporném režimu.

3.1 Porovnání s klasickým Bluetooth

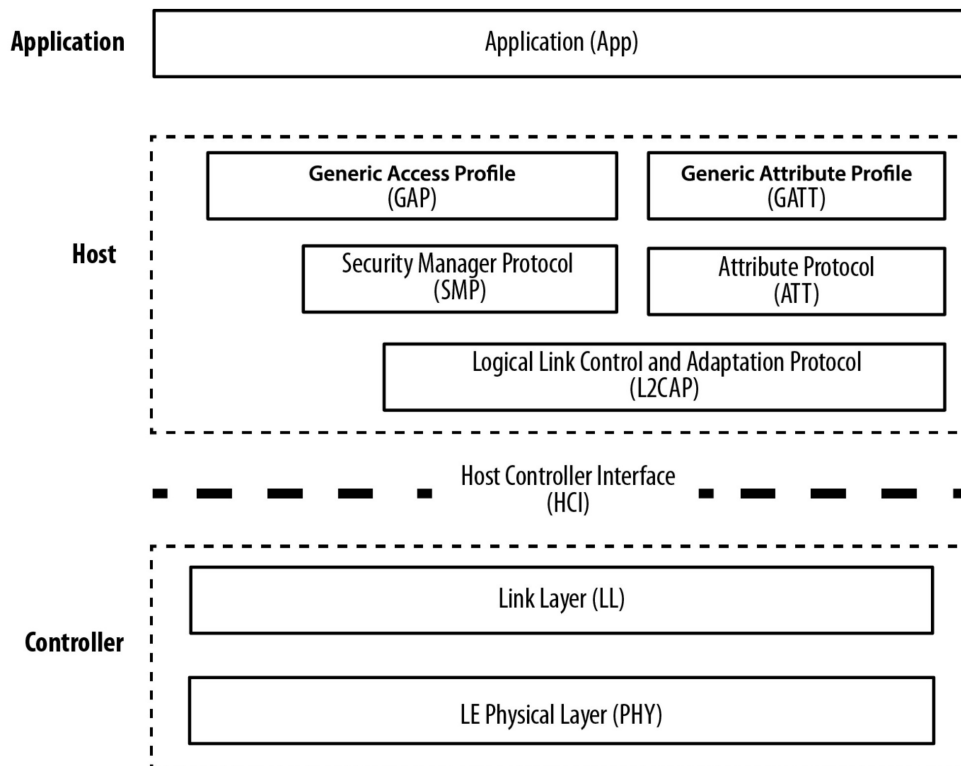
Bluetooth Low Energy je sice součástí specifikace Bluetooth 4.0, ale není zaručeno, že zařízení s touto specifikací je BLE kompatibilní. Specifikace totiž umožňuje implementaci pouze klasického Bluetooth a nebo jeho kombinaci s BLE. Schopnost zařízení používat BLE je tedy dána výrobcem.

Stejně jako klasický Bluetooth, využívá BLE bezlicenční ISM frekvenční pásmo 2.400 - 2.4835 GHz. Bluetooth LE používá kanály o šířce 2 MHz. Do rozsahu 2.400 až 2.4835 GHz se jich vejde 40, tedy polovina počtu klasického Bluetooth, který používá kanály se šířkou 1 MHz. Bit-rate komunikace je 1 Mbit/s. S představením verze 5 byl označen jako 1M PHY. Bluetooth 5 dále přidal 2M PHY, s rychlostí 2 Mbit/s a Coded PHY se sníženou rychlostí, ale umožňující komunikaci na delší vzdálenosti.

Dosah přenosu se nedá přesně specifikovat, protože je závislý na mnoha faktorech. Hlavními faktory jsou prostředí, překážky, které musí signál překonat, nebo mód, ve kterém komunikace probíhá. Na oficiálních stránkách Bluetooth SIG se pro PHY 1 uvádí i více než 100 m a při použití Coded PHY je možno dosáhnout vzdálenosti přesahující 1 km [4].

3.2 Architektura

Tato kapitola popisuje architekturu sady protokolů Bluetooth stacku od nejnižší vrstvy reprezentující fyzickou anténu až po nejvyšší abstrakční vrstvy, ke kterým přistupují aplikace. Hierarchie jednotlivých vrstev je znázorněna na obrázku 3.1. Informace v této kapitole jsou převzaty z [21].



Obrázek 3.1: Hierarchické zobrazení komponent architektury tvořící specifikaci Bluetooth Low Energy [21].

LE Physical Layer (PHY)

Fyzická vrstva obstarává analogovou komunikaci a její převod do digitální podoby. PHY může být nastavena do několika režimů ovlivňujících přenosovou rychlost a maximální vzdálenost [4], [1].

- **1M PHY** - původní komunikační režim verze Bluetooth 4.0. Posílá data rychlostí 1MB/s.
- **2M PHY** - režim představený ve specifikaci 5.0, rozšiřuje možnosti komunikace o režim s dvojnásobnou rychlostí 2 MB/s. Toho je dosaženo zvýšením hustoty posílaných znaků. Zkrácením doby odesílání stejně velkých dat je dosaženo menší spotřeby. Nevýhodou je větší náchylnost k chybám při přenosu a kratší dosah.
- **Coded PHY** - tento režim byl přidán spolu s 2M PHY v Bluetooth 5.0. Jeho přínosem je zvětšení vzdálenosti, na kterou je možné komunikovat. Dosaženo je toho tím, že 1 bit přenášených dat je kódován do několika (2 nebo 8) fyzicky posílaných bitů. Podle on-line nástroje na odhad dosahu Bluetooth¹ lze takto komunikovat na vzdálenosti přesahující 1 kilometr.

¹Webová aplikace pro výpočet dosahu Bluetooth vysílání v několika rozličných prostředích a při použití různých nastavení <https://www.bluetooth.com/learn-about-bluetooth/key-attributes/range/>

Link Layer (LL)

Linková vrstva přímo komunikuje s fyzickou vrstvou a poskytuje vyšším vrstvám abstraktní rozhraní pro komunikaci s fyzickou vrstvou. Tato vrstva definuje několik rolí, které mohou zařízení při komunikaci zastávat. Role advertiser a scanner jsou role sloužící k navázání spojení, případně slouží ke všesměrové komunikaci. Master a slave jsou role zařízení, které už spojení navázaly a komunikují výhradně mezi sebou.

- **Advertiser** - zařízení, které vysílá advertising pakety.
- **Scanner** - zařízení, hledající příchozí advertise pakety.
- **Master** - zařízení, které inicializuje spojení a později jej řídí.
- **Slave** - zařízení, které přijme žádost o spojení s master zařízením.

Identifikátorem Bluetooth zařízení je 48 bitová adresa Bluetooth. Při komunikaci může být použita buď veřejná, ta je fixní a nastavená výrobcem a registrovaná u IEEE nebo náhodná adresa, ta je pseudonáhodně vygenerovaná zařízením.

Host Controller Interface (HCI)

Toto rozhraní zprostředkovává komunikaci mezi Controllerem, který se skládá z PHY a LL, a vyššími vrstvami. Controller musí řešit pevně dané časování fyzické vrstvy tak, aby zařízení mohlo bezproblémově komunikovat s ostatními zařízeními s ohledem na splnění specifikace. Vyšší vrstvy v Host se starají o složitější operace, takže bývají řízeny procesorem, zatímco Controller bývá implementován na čipu poskytujícím CPU přístup k Bluetooth jako I/O operace.

Logical Link Control and Adaptation Protocol (L2CAP)

Tato vrstva zapouzdřuje protokoly z vyšších vrstev, také slouží k rozdělování a opětovnému spojování velkých paketů předaných vyššími vrstvami. Velikost fyzicky odeslaného paketu může být 27 bytů, proto jsou data rozdělována na 23 bytové části, je potřeba počítat ještě se 4 byty tvořícími hlavičku.

Security Manager Protocol (SMP)

Security Manager (SMP) je protokol, který má na starosti párování. Poskytuje možnost generovat a vyměňovat šifrovací klíče. Může také skrýt veřejnou Bluetooth adresu spojenou s identitou zařízení a používat náhodně generované adresy.

Generic Access Profile (GAP)

GAP je nejvyšší kontrolní vrstva, jejímž úkolem je umožnit vyšším vrstvám vysílat data, objevovat zařízení, navazovat a udržovat spojení. Většina API ovládajících Bluetooth poskytuje přístup pouze ke GAP.

Tento profil definuje následující role, které během komunikace zařízení používají.

- **Broadcaster** - periodicky rozesílá “advertising“ pakety a očekává připojení jiného zařízení (připojené zařízení vstoupí do role central a broadcaster se stane zařízením typu peripheral).

- **Observer** - pozorovatel pouze hledá advertising zařízení, čte data která sdílejí, ale nepřipojuje se k nim.
- **Peripheral** - “advertising“ pakety, které rozesílá periferní zařízení mají nastavený příznak connectable, to znamená, že umožňuje ostatním zařízením, aby se k němu připojila a četla charakteristiky, které poskytuje.
- **Central** - opakovaně skenuje na advertising frekvencích a inicializuje připojení k periferním zařízením. Když je spojení vytvořeno, určí časování komunikačních oken, následně čte a zapisuje charakteristiky peripheral zařízení.

Attribute Protocol (ATT)

Attribute protocol je bezstavový klient-server protokol založený na vystavování dat v attributech. Atributy jsou nejmenší datové entity používané pro výměnu dat v BLE. Atribut se skládá z několika částí:

- **Attribute Handle** - unikátní 16 bitový identifikátor umožňující adresování jednotlivých atributů.
- **Attribute Type** - UUID určující datový typ uložený v hodnotě.
- **Attribute Permission** - specifikuje jaké operace je možné provádět s konkrétním atributem (jedná se například o čtení, zápis, notifikaci, zašifrování apod.).
- **Attribute Value** - vlastní hodnota atributu, data v něm uložená mohou být jakéhokoliv typu, jedná se v podstatě o bytové pole.

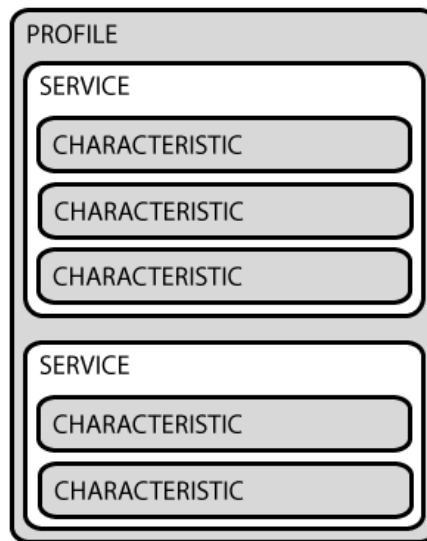
Klient a server spolu komunikují způsobem definovaným ve specifikaci BLE. Podle ní může být zařízení buď klient nebo server, ale může zastávat i obě role.

Server zpřístupňuje služby (services) a určuje jejich obsah a vlastnosti. Při změně hodnoty může notifikovat klienta nebo mu povolit zápis, tím umožní serveru oboustrannou komunikaci. Serverem obvykle bývají zařízení, která jsou zdrojem dat ke kterým chceme získat přístup. Příkladem mohou být chytré hodinky, smart beacons, chytré teploměry apod.

Klient inicializuje spojení a čte charakteristiky ze serveru. Typicky tuto roli zastávají zařízení, které data zobrazují uživateli jako jsou mobilní telefony, tablety nebo PC.

Generic Attribute Profile (GATT)

GATT má na starosti komunikaci mezi zařízeními, které už spojení navázaly pomocí GAP. Pro výměnu dat používá služby a charakteristiky. Ty jsou uloženy v jednoduché vyhledávací tabulce, která je součástí protokolu ATT (Attribute Protocol), a jsou identifikovány pomocí 16 bitových identifikátorů UUID. GATT také využívá role klient a serveru definované v ATT.



Obrázek 3.2: Struktura uložení dat [20].

Universally unique identifier (UUID) je univerzální identifikátor o délce 128 bitů (16 bytů). Jedná se o číslo, u kterého je garantováno, že je s vysokou pravděpodobností unikátní. UUID jsou používány i jinde než v Bluetooth protokolu. Za účelem zefektivnění přenosu dat definuje Bluetooth SIG zkrácené UUID o délce 16 bitů (2 byty) a 32 bitů (4 byty), tím lze získat místo při posílání obecně používaných služeb a charakteristik. Takovými charakteristikami jsou např. charakteristiky pro zjištění úrovně baterie nebo přenos informací o srdečním tepu. Zkrácené UUID mohou být používány pouze jako UUID přidělené ve specifikaci Bluetooth SIG². Takto zkrácená verze je poté doplněna o základní UUID Bluetooth:

xxxxxxxx-0000-1000-8000-00805F9B34FB

Service (Služba) Služba sdružuje podobné atributy do jedné skupiny. Atributy služby jsou většinou charakteristiky, mohou to ale být i jiné služby. Hodnota služby se skládá z UUID a hodnoty jakéhokoli typu [13].

Characteristic (Charakteristika) Charakteristika zaobaluje samotná data, která jsou přes Bluetooth posílána. Obsahuje vlastní hodnotu a může obsahovat deskriptory [13].

Descriptor (Deskriptor) Deskriptor poskytuje klientovi metadata o charakteristice. Může například obsahovat pro člověka srozumitelný popis charakteristiky, její jednotku nebo rozsah hodnot, kterých může nabývat [13].

3.3 Princip komunikace

Komunikace pomocí Bluetooth probíhá typicky nejprve pomocí kombinace advertisementu a skenování, které umožňují všesměrovou komunikaci s okolními zařízeními. Tato kombinace

²Seznam 16bitových čísel přidělených Bluetooth SIG je dostupný z <https://www.bluetooth.com/specifications/assigned-numbers/>

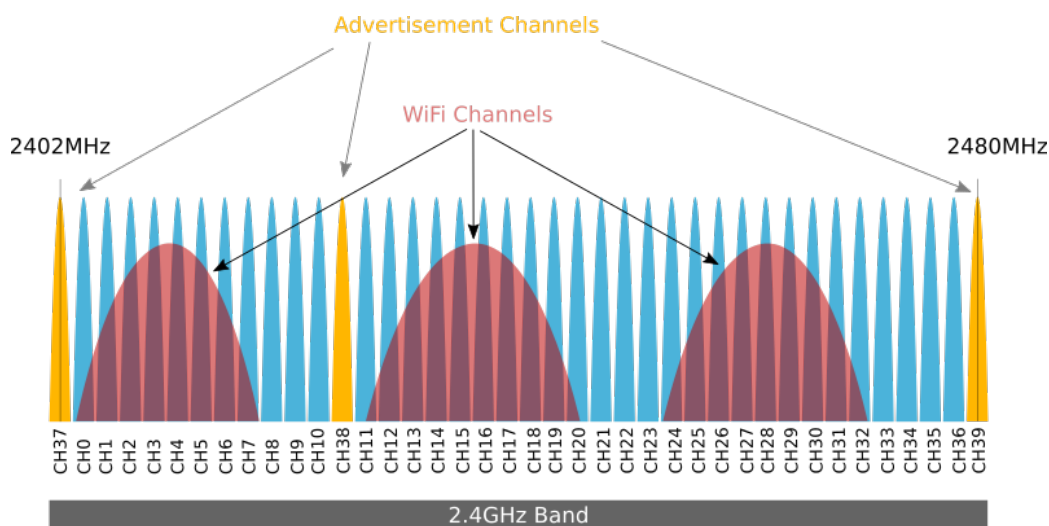
slouží primárně k navázání a vytvoření spojení mezi dvěma konkrétními zařízeními. Po vytvoření spojení už komunikují přímo a komunikace může být zabezpečena šifrováním.

Advertising (Propagace)

Jedním ze dvou druhů paketů v BLE je Advertisement paket. Slouží k předávání informací mezi nepřipojenými zařízeními. To může být potřeba před vytvořením spojení, kdy zařízení propaguje svoji existenci. Informace, které tak zařízení poskytuje, použije centrální zařízení k výběru zařízení, ke kterému se chce připojit.

Advertising paket může obsahovat až 31 bytů dat spolu s informacemi v hlavičce. Paket může obsahovat jméno zařízení, informace o výrobci, UUID identifikující služby, které zařízení poskytuje nebo omezené množství dat. Dále obsahuje příznak informující jestli je možné se k zařízení připojit nebo jestli paket obsahuje nějaká data.

Jelikož je 31 bytů omezující a existují zařízení komunikující výhradně pomocí advertise paketů, např. Bluetooth beacons, specifikace Bluetooth 5.0 zvyšuje maximální velikost advertise paketu na 255 bytů pomocí periodického advertisementu [16].



Obrázek 3.3: Kanály používané Bluetooth Low Energy [21].

Pro advertisement jsou vyčleněny 3 ze 40 kanálů, na kterých Bluetooth operuje. Jedná se o kanály 37, 38 a 39, jak je možné vidět na obrázku 3.3. Jejich frekvence je zvolena tak, aby byla co nejmenší interference mezi nimi a nepoužívanějšími kanály Wi-Fi (1, 6 a 11), která používá stejnou část elektromagnetického spektra jako Bluetooth.

Aby se co nejvíce minimalizovala možnost, že bude advertisement interferovat s jiným vysíláním na stejné frekvenci, probíhá advertisement postupně na všech třech kanálech.

Scanning (Skenování)

Vysílání advertising paketů a skenování nemohou být nijak synchronizovány, proto úspěšný příjem paketu nastane pouze když se překryjí intervaly, po který jedno zařízení vysílá a druhé skenuje. Skenování proto probíhá stejně jako vysílání střídavě na všech kanálech.

Existují dva druhy skenování:

- **Pasivní skenování** - skener pouze poslouchá příchozí pakety.

- **Aktivní skenování** - po přijetí advertise paketu posílá skener odpověď a na ni může advertiser reagovat posláním paketu obsahujícím další informace.

Příchozí advertise pakety mohou být filtrovány. Filtrování lze nastavit např. podle adresy zařízení, nebo informací které o sobě poskytuje jako jméno nebo sdílená data.

Connect (Připojení)

Aby bylo možné vytvořit spojení, musí nejprve proběhnout komunikace pomocí advertise paketů vysílaných slave zařízeníem a jejich skenování masterem. Po té, co master objeví slave zařízení, se kterým se chce spárovat, pošle mu connection request paket. Po přijetí odpovědi od slave je ustaveno spojení.

Request paket obsahuje informace o tom, jakým způsobem budou při komunikaci měněny kanály, intervaly kdy budou zařízení komunikovat a kdy mohou být přepnuty do úsporného režimu. Dále obsahuje informace o tom, kolik komunikačních oken může být vynecháno, než bude spojení považováno za neplatné.

Při spojení pomocí Bluetooth je možné použít šifrování. Existují dva způsoby šifrovaného spojení, lišící se délkou jeho trvání:

- **Pairing** - párování je způsob, jak vytvořit zabezpečené spojení. Toto zabezpečení trvá pouze po dobu jednoho spojení. Po jeho ukončení musí zařízení opět projít procesem párování, pokud chtějí opět šifrovaně komunikovat.
- **Bonding** - bonding je proces, který proběhne stejně jako párování, ale navíc dojde k výměně permanentních šifrovacích klíčů, které slouží při opětovném navazování šifrovaného spojení.

Kapitola 4

Bluetooth Mesh

BLE Mesh je softwarová nadstavba nad Bluetooth umožňující komunikaci typu many-to-many. Umožňuje tedy posílání informací mezi jakýmkoliv dvěma zařízeními připojenými k síti. Zařízení používající Mesh často kladou důraz na nízkou spotřebu, protože jsou napájena z baterie a musí vydržet fungovat řádově několik měsíců až let. Jedná se hlavně o čidla, světla, vypínače. Využití najde ale i při orientaci ve vnitřních prostorech rozsáhlých budov nebo při vytváření síťové infrastruktury na místech zasažených katastrofou. Specifikace Bluetooth SIG byla vydána v roce 2017. Svou implementaci BLE Mesh má mnoho výrobců jako například Qualcomm, Samsung, Nordic Semiconductor a další. Informace o Bluetooth Mesh vycházejí z [14].

4.1 Vznik

Doposud diskutované technologie Bluetooth a Bluetooth Low Energy pracují s klasickým paradigmatem point-to-point, tedy komunikace jeden na jednoho. A i když umožňují vytváření menších sítí typu piconet, jedná se pouze o několik periferních zařízení připojených k jednomu centrálnímu. Toto řešení nevyhovovalo použití v Internet of Things. Ten vyžaduje, aby spolu mohla komunikovat i dvě zařízení, která jsou od sebe vzdálena tak, že spolu nemohou komunikovat přímo.

Z toho důvodu vzniklo několik různých řešení komerčních a také pro účely akademických studií, fungujících různými způsoby. Tato řešení však nejsou vzájemně kompatibilní. Aby byla zaručena standardizace Bluetooth SIG představila své řešení - Bluetooth Mesh [2].

4.2 Princip komunikace

Komunikace v Bluetooth Mesh probíhá pomocí skenování a vysílání advertise paketů. Tímto způsobem jsou zprávy posílány a přeposílány dále, což zvětšuje dosah původní zprávy. Ke směrování paketů je použita metoda *managed-flooding*.

Flooding (záplavová) metoda využívá při průchodu paketu sítí od odesilatele k příjemci všech možných cest. Uzel sítě totiž při přijetí paketu, který není určen pro něj, jej přepoše všem svým sousedům.

Aby bylo zabráněno opakovanému přeposílání stejné zprávy, a tím i zahlcení sítě, jsou použity dvě metody.

- **time-to-live** - tato metoda se zakládá na tom, že při vytvoření paketu je mu přidělena hodnota *time-to-live*. Ta je s každým preposláním dekrementována. Po snížení hodnoty na nulu už paket není preposlán dál.
- **paket caching** - tato metoda ukládá přijaté pakety do seznamu v každém zařízení. Když zařízení obdrží paket, který už má uložený, znovu jej nepřeposílá.

Předávání dat v rámci Bluetooth Mesh sítě probíhá na základě tzv. *publish-subscribe* paradigmatu. Uzly generující zprávy je publikují na některou z adres. Zařízení mající zájem o tyto informace se přihlásí k odběru z této adresy.

Aby byla zajištěna co nejvyšší šance přijetí všech paketů, je nastavena délka intervalu skenování na 100 % času. To znamená, že zařízení neustále skenuje. Skenování přerušuje pouze pokud samo odesílá nějaký paket.

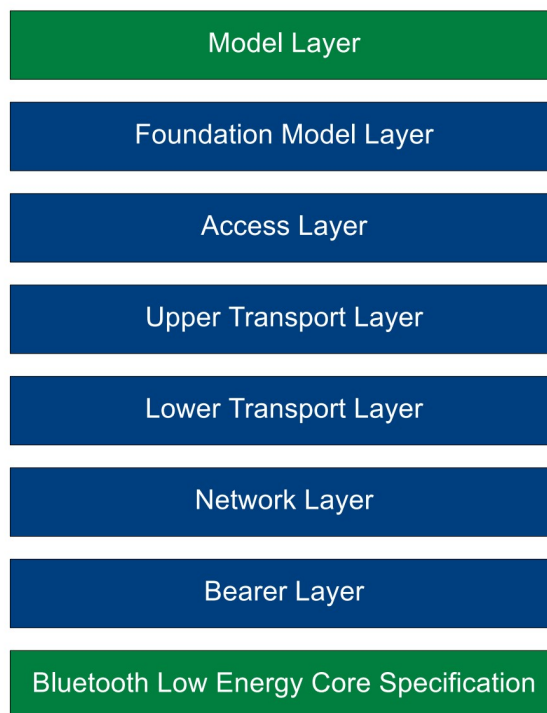
Adresy

- **Unicastová adresa** - každému zařízení je při připojení k síti přidělena jedna z 32 767 číselných adres, která ho jednoznačně identifikuje v rámci této sítě.
- **Virtuální adresa** - tato adresa je multicastová a může reprezentovat několik prvků na několika zařízeních. Logicky zastupuje UUID používané k identifikaci služeb a charakteristik. Přesto, že je UUID 128 bitová hodnota, je pro účely adresování hashována na jednu ze 16 384 hodnot.
- **Skupinová adresa** - jedná se také o multicastovou adresu. Reprezentuje jeden nebo více uzlů.

4.3 Architektura

Základem komunikace Mesh je Bluetooth Low Energy. Nad ním jsou postaveny další vrstvy umožňující komunikaci více zařízení. Architektura Mesh je znázorněna na obrázku 4.1. Bluetooth Mesh používá ke komunikaci tři advertising kanály Bluetooth Low Energy. Použití pouze těchto tří kanálů je dáno tím, že Mesh využívá jen advertise pakety. Zařízení mezi sebou nenavazují spojení nebo párování jako při komunikaci pomocí Bluetooth Classic a Bluetooth Low Energy. Pakety používané v Mesh mají nastavené příznaky nepřipojitelný a neskenovatelný.

Většina BLE zařízení, nemá implementovány Mesh protokoly. Proto Mesh podporuje režim zpětné kompatibility. Tento režim umožní připojit se k síti i BLE zařízením. Funguje tak, že některé z Mesh zařízení zpřístupní svůj GATT server, ke kterému se BLE zařízení připojí [14]. To znamená, že se k síti může připojit například chytrý telefon, který slouží jako ovladač chytrého osvětlení, nebo dokáže získat data ze senzorů, které jsou připojeny k síti, a zobrazit je uživateli. Popis architektury vychází z [14].



Obrázek 4.1: Bluetooth Mesh stack, převzato z [14].

Model layer

Modelová vrstva je nejvyšší vrstvou technologie Mesh definuje tzv. modely.

Model reprezentuje standardizované chování zařízení připojeného k síti. Jeden uzel může obsahovat více modelů. Model definuje množinu stavů, ve kterých se může nacházet. Dále obsahuje množinu zpráv, kterými na tyto stavy reaguje. Svůj model může mít např. žárovka nebo vypínač.

Modely se dělí na:

- **Server model** - tento model definuje množinu zpráv, které může server posílat a přijímat. Dále definuje chování, kterým reaguje na jednotlivé zprávy.
- **Client model** - klientský model obsahuje zprávy, kterými může komunikovat se serverem a neobsahuje žádný stav.
- **Control model** - kontrolní model může obsahovat server model i klient model. Slouží ke komunikaci se serverem i s klienty.

Foundation Model layer

Základová modelová vrstva definuje stavy, zprávy a modely sloužící ke konfiguraci a řízení mesh sítě.

Access layer

Přístupová vrstva definuje formát dat, řídí šifrování a dešifrování ve vyšší transportní vrstvě.

Upper transport layer

Vyšší transportní vrstva slouží k šifrování, dešifrování a autentizaci dat aplikací.

Lower transport layer

Nižší transportní vrstva má na starosti rozdělování a spojování zpráv.

Network layer

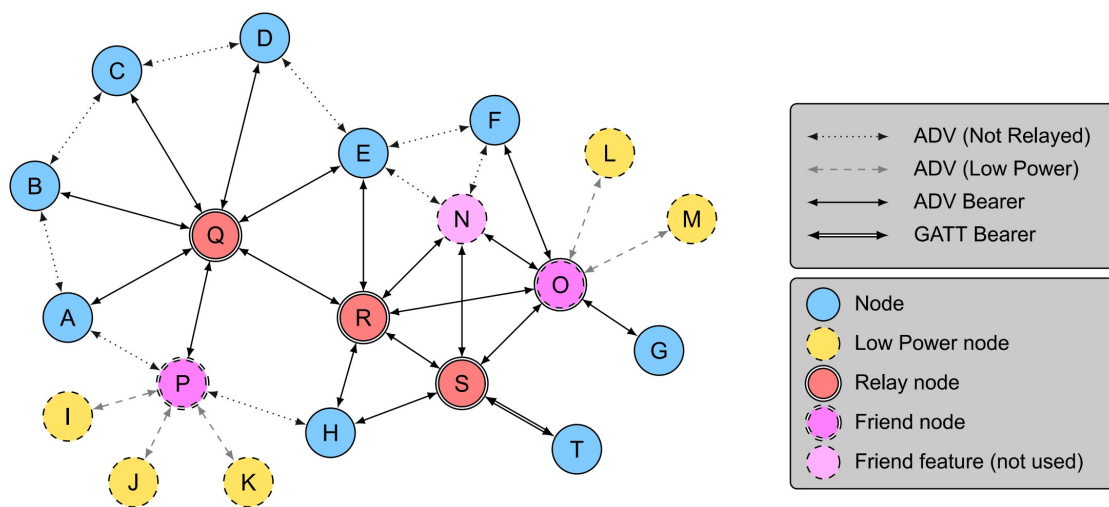
Síťová vrstva určuje, jak budou zprávy směrovány cílovým zařízením.

Bearer layer

Doručovatelská vrstva definuje, jak bude zpráva odeslána mezi dvěma uzly sítě.

4.4 Topologie

Už z názvu Bluetooth Mesh vyplývá, že používá síťovou neboli smíšenou topologii. Mesh topologie [15] je taková topologie, ve které je každý komunikující uzel propojen se všemi ostatními uzly. Bluetooth Mesh ale nevyžaduje přímé spojení mezi všemi uzly. Jedná se tedy spíše o částečnou mesh topologii, protože hlavní motivací ke vzniku Bluetooth Mesh bylo, aby spolu mohla komunikovat i zařízení, která nejsou přímo připojena.



Obrázek 4.2: Topologie BLE Mesh, převzato z [14].

Na obrázku 4.2 jsou barevně rozlišeny různé druhy uzlů. Zavedení těchto specifických druhů uzlů omezuje množství dat, které se sítí pohybuje, nebo umožňuje připojení zařízení, která by jinak nebyla schopna fungovat v Mesh jako plnohodnotné uzly z důvodu energetické náročnosti Mesh [14].

- **Relay node** - tento druh uzlu existuje, aby bylo omezeno množství dat posílaných po síti. Pokud síť podporuje relay feature, jsou zvoleny uzly, které mohou jako jediné přeposílat přijaté zprávy. To šetří prostor v přenosovém pásmu a zároveň výpočetní

výkon potřebný ke zpracování části paketů, které zařízení přijalo už v minulosti. V obrázku jsou tyto uzly označeny červenou barvou.

- **Node** - Modrou barvou jsou označeny uzly fungující v normálním režimu. Pokud je aktivní relay feature, nepřeposílají přijaté zprávy.
- **Low power node** - některá zařízení v síti nemohou plnit roli plnohodnotného uzlu, protože jsou napájena z baterie. Uzly Mesh používají nepřetržité skenování, aby mohly správně komunikovat. Nízkospotřebová zařízení se připojí k jinému uzlu, který je schopný plnohodnotné Mesh komunikace a slouží jako jejich prostředník pro komunikaci s ostatními uzly sítě. Uzel, ke kterému se připojí, se stává jejich přátelským uzlem.
- **Friend node** - přátelský uzel plní funkci prostředníka mezi low power uzlem a zbytkem sítě. Jeho úkolem je ukládat zprávy určené uzlu low power. Přeposílá mu je ve specifikovaných intervalech, takže uzel low power nemusí nepřetržitě skenovat a šetří tak energii.

Provisioning

Provisioning je proces přidávání zařízení do sítě. Tento proces má na starosti poskytovatel (Provisioner). Ten poskytne zařízení informace potřebné pro jeho připojení k síti. K připojení je potřeba získat zejména síťový klíč a unicastovou adresu. Poskytovatelem typicky bývá mobilní telefon nebo počítač.

4.5 Neoficiální Bluetooth síť

Jak bylo zmíněno již v úvodu této kapitoly Bluetooth Mesh oficiální specifikace není jediným řešením, které používá Bluetooth ke komunikaci více zařízení. Přehled sítí založených na komunikaci pomocí Bluetooth je představen v [6]. Tato řešení se mezi sebou liší zejména způsobem propagování paketů v síti a použitou verzí Bluetooth.

Některá řešení využívají záplavový způsob propagace paketů stejně jako Bluetooth Mesh a aplikují různé algoritmy sloužící k omezení počtu přeposílání paketů.

- **Trickle algoritmus** - při použití algoritmu stékání, zařízení vysílá data v náhodných intervalech. Intervaly se postupně prodlužují, pokud jsou data v síti konzistentní. Po přijetí informací, která konzistentní není, aktualizuje svou informaci a zkrátí interval, ve kterém vysílá, na původní hodnotu [17].
- **Gossiping** - na základě tohoto algoritmu je přijatý paket předán náhodnému sousedovi. Při šíření informace se tak algoritmus spoléhá na to, že po několika takovýchto předáních se informace rozšíří do celé sítě [19].

Další skupina využívá směrování. Řešení využívající směrování nejčastěji organizují uzly svých sítí do acyklického grafu. Tyto sítě používají směrování statické nebo dynamické.

- **Statické směrování** - síť používající statické směrování organizují uzly sítě do stromové topologie. Statické sítě jsou náchylné k výpadku při selhání jednoho uzlu. Tomu se lze částečně vyhnout výběrem alternativního rodiče, kterému jsou posílána data, když rodičovský uzel není dostupný.

- **Dynamické směrování** - při dynamickém směrování jsou využity dva základní přístupy. Prvním je on-demand směrování, kdy je nejvýhodnější cesta mezi zdrojem a cílem hledána při každém zasílání paketu. Druhým je vytvoření acyklického stromového grafu a směrovací tabulky obsahující přinejmenším informace o jeho bezprostředním okolí, tedy rodičovském uzlu a potomcích.

Kapitola 5

Prostředí Android

Android API neposkytuje kompletní přístup k Bluetooth stacku, navíc upravuje některé funkce s ohledem na zabezpečení operačního systému. Tato sekce se zabývá poskytnutým rozhraním pro přístup k Bluetooth Low Energy, zohledněním toho co Android umožňuje a tím na co je brán při implementaci zřetel. Následuje popis služeb, které je potřeba využít, aby mohla aplikace běžet na pozadí.

5.1 Podpora Bluetooth Low Energy

Softwarová podpora Bluetooth Low Energy byla přidána do verze systému 4.3 Jelly Bean (API level 18) v roce 2013, tehdy ještě pod jménem Bluetooth Smart [7]. Jednalo se pouze o omezenou podporu umožňující komunikaci v roli centrálního zařízení. Rozhraní pro práci s BLE bylo dále rozšířeno o roli periferního zařízení ve verzi API 21, tedy Androidu verze 5.0 Lollipop [8]. Toto rozšíření dále zahrnuje například vylepšení skenování, propagaci nebo nastavení PHY [12].

5.2 Rozhraní pro práci s Bluetooth Low Energy

Bluetooth rozhraní je v Androidu reprezentována třídou `BluetoothAdapter`, který pracuje s klasickým Bluetooth i BLE. Pro celý systém existuje jedna instance této třídy, všechny aplikace tedy pracují se stejným objektem. Pokud zařízení nepodporuje Bluetooth při pokusu o získání `BluetoothAdapter` vrací `getSystemService()` `null` [9] [22].

Rozhraní pro přístup k Bluetooth je asynchronní. To znamená, že rozhraní může být předán požadavek, který ho ale neblokuje pro ostatní. Tímto způsobem může být předáno několik požadavků najednou a při dokončení jednotlivých operací zavolá Bluetooth některou z tzv. callback funkcí, jejíž výběr závisí na výsledku dané operace. Zpracování výsledku operace potom probíhá v této callback funkci.

Podpora skenování

K hledání okolních zařízení je třeba začít skenovat. O to se stará třída `BluetoothLeScanner`.

Při skenování je často potřeba najít zařízení poskytující konkrétní služby. Proto se dají výsledky skenování filtrovat ještě před tím než jsou předány aplikaci. Filtrování lze provádět podle:

- MAC adresy zařízení

- jména zařízení
- service UUID identifikujícího poskytované služby
- dat některé služby
- informací o výrobci zařízení

Kromě filtrů se dají nastavit také další parametry jako je např. `scanMode`, který určuje režim skenování a dále `CallbackType`, který určuje kolik výsledků skenování bude posláno aplikaci. Správná volba `scanMode` je důležitá, protože slouží k nastavení intenzity skenování, což ovlivňuje spotřebu baterie. Je třeba zvolit správný kompromis mezi spotřebou a latencí. Bližší informace o tom, jak tyto režimy fungují, oficiální dokumentace neposkytuje. Při hledání podrobnějších informací jsem narazil na tento dotaz na Stack Overflow¹ a odpověď odkazující se přímo do zdrojového kódu Android Opensource Project².

- `SCAN_MODE_BALANCED` - skenování v balancovaném režimu znamená, že sken probíhá 1 vteřinu a 4 vteřiny je nečinný.
- `SCAN_MODE_LOW_LATENCY` - tento režim má nejvyšší spotřebu. Jeho použití znamená důraz na co nejmenší latenci, tedy co nejrychlejší objevení všech zařízení v okolí. Pracuje v intervalu 4 vteřiny skenování a 4 vteřiny je vypnutý.
- `SCAN_MODE_LOW_POWER` - provádí BLE sken v režimu s nízkou spotřebou. Samotný sken trvá 0,5 sekundy a je vypnutý následujících 5 sekund. Periferní zařízení musí vysílat advertisement často, aby mohlo být v tomto režimu spolehlivě objeveno. Tento mód je zvolen jako defaultní.
- `SCAN_MODE_OPPORTUNISTIC` - aplikace získává pouze výsledky skenu spuštěného jinou aplikací, sama sken nespouští.

Advertising

O proces advertisementu se stará třída `BluetoothLeAdvertiser`. U advertisementu jsou kromě nastavení vlastních dat jako nejdůležitější parametry `Advertise Mode` a `Advertise TX Power`. Ty lze nastavit pouze na některou předpřipravenou hodnotu.

- `ADVERTISE_MODE_LOW_LATENCY` - advertising interval je 100 ms.
- `ADVERTISE_MODE_BALANCED` - advertising interval je 250 ms.
- `ADVERTISE_MODE_LOW_POWER` - advertising interval je 1000 ms.

TX power lze nastavit na `ADVERTISE_TX_POWER_HIGH`, `ADVERTISE_TX_POWER_MEDIUM`, `ADVERTISE_TX_POWER_LOW`, `ADVERTISE_TX_POWER_ULTRA_LOW`.

¹<https://stackoverflow.com/questions/48686074/android-ble-scan-mode-setting-interval>

²Zdrojový soubor `ScanManager.java` s definicí konstant používaných při ovládní skenování Bluetooth <https://android.googlesource.com/platform/packages/apps/Bluetooth/+master/src/com/android/bluetooth/gatt/ScanManager.java>

Omezení dané operačním systémem Android

Operační systém Android zavádí několik omezení. Protože skenování spotřebovává energii, je možné ho provádět jen po omezenou dobu, kterou lze určit nastavením limitu. Pokud není sken ukončen po 30 minutách, ukončí jej systém a přepne skenování do režimu `SCAN_MODE_OPPORTUNISTIC`. Omezen je i počet spuštěných skenů v určitém časovém intervalu, a to na 5 spuštěných skenů za dobu 30 sekund. Pokud je tento limit překročen, systém další sken následujících 30 vteřin nespustí. Dalším omezením ze strany Androidu je, že skenování, které není spuštěno s nastavenými filtry, je ukončeno pokaždé, když je vypnuta obrazovka. Tento problém se dá obejít vytvořením prázdného filtru, který propustí všechny příchozí pakety [22].

5.3 Oprávnění

Ke správnému fungování vyžaduje aplikace několik oprávnění, to se týká přístupu k Bluetooth [10]. Aby aplikace mohla vytvořit spojení a posílat data pomocí Bluetooth, je potřeba požádat o oprávnění `BLUETOOTH`. Ke skenování Bluetooth zařízení v okolí a manipulaci nastavení slouží oprávnění `BLUETOOTH ADMIN`. Aplikace potřebuje také oprávnění `ACCESS_COARSE_LOCATION`, protože viditelná zařízení mohou prozradit polohu uživatele.

Žádost o oprávnění je provedena následující deklarací v souboru `AndroidManifest.xml`:

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

Oprávnění přístupu k poloze v telefonech s verzí android 10 a výše je navíc potřeba manuálně nastavit na „Povolit vždy“ v systémovém nastavení. To má uživatele přimět k zamyšlení, jestli chce dané oprávnění opravdu udělit. Stejně je potřeba v nastavení vypnout optimalizaci spotřeby baterie, jejímž úkolem je ukončovat služby běžící na pozadí. Někteří výrobci tuto službu navíc implementují tak, že je velice agresivní. Mohlo by se tedy stát, že by aplikace byla vypnuta systémem.

5.4 Služby

Služby (Services) slouží jako prostředek pro vykonávání operací na pozadí. Jakmile je služba spuštěna, běží na pozadí po dobu, než ji ukončí systém, i když uživatel přepne do jiné aplikace. Jelikož běží na pozadí, neposkytuje služba uživatelské rozhraní. Služby typicky využívají aplikace jako přehrávače hudby, aplikace komunikující přes síť nebo aplikace pracující s I/O.

Existují tři základní druhy služeb:

- **Služba na popředí** - má trochu zavádějící název, ale stále slouží k provádění operací, které nevyžadují přímou interakci s uživatelem. Používají ji aplikace, které uživatel přímo vidí jako např. přehrávání audia. Tyto služby jsou aktivní i když uživatel neinteraguje s aplikací. Služby na popředí musí zobrazovat notifikaci, aby bylo uživateli jasné, že běží. Notifikace může také obsahovat tlačítka pro základní interakci s aplikací.
- **Služba na pozadí** - provádí operace, kterých si uživatel přímo nevšimne jako služba uklízející úložiště své aplikace.

- **Vázaná služba** - umožňuje ostatním komponentám interakci typu klient-server. Komponenty jí mohou posílat požadavky a získávat výsledky. Vázaná služba běží tak dlouho, dokud k ní jsou vázané další komponenty. Po rozvázání všech komponent je tato služba systémem ukončena.

Službu lze spustit dvěma způsoby. Voláním funkce `startService()` se spouští tzv. spuštěná služba na popředí nebo na pozadí. Ta je po svém vytvoření nezávislá na komponentě, která ji spustila a ukončuje buď sama nebo jinou komponentou voláním `stopService()`. Vázaná služba je spuštěna pokud se k ní některá komponenta „přiváže“ pomocí `bindService()`. Ukončena je po zavolání `unbindService()` všemi vázanými komponentami [11].

Kapitola 6

Návrh řešení

Tato kapitola se zabývá kompletním návrhem aplikace s ohledem na požadavky na jeho fungování. Nejprve je třeba zvolit technologii pro komunikaci, poté navrhnout komunikační protokol. Dále je potřeba zvolit jakým způsobem budou členové sledováni a jakým způsobem a kdy budou upozorněni ostatní členové skupiny v případě něčí ztráty.

6.1 Celkový princip

Skupinu vytvoří jeden člen, organizátor, jehož chování bude mít podobný charakter jako zařízení provisioner definované v Bluetooth Mesh sítích. K tomuto zařízení se připojí všichni členové skupiny a získají od něj identifikátory a poskytovatel si bude zároveň vytvářet seznam připojených zařízení. Jakmile se připojí zařízení všech, kteří chtějí být členy skupiny, může organizátor ukončit tuto úvodní fázi a předat svůj seznam připojených členů skupiny všem už připojeným zařízením. Následně se zařízení mohou začít organizovat do sítě.

Organizace proběhne tak, že všechna zařízení začnou vysílat své identifikátory a získávat identifikátory ostatních. Při získání informace vysílané některým z ostatních zařízení si zařízení uloží informaci o tom, že je toto zařízení v jeho dosahu. Tuto činnost budou provádět po určitou dobu a následně zpracují získané informace. Poté začnou opět vysílat své identifikátory a přidají i informaci o zařízeních objevených v průběhu předchozího komunikačního intervalu.

Při přijetí zprávy obsahující informaci i o jiném zařízení, než je to, které je autorem přijaté zprávy, příjemce zkontroluje informaci o daném zařízení ve své tabulce. Pokud v tabulce nemá záznam s potřebnými informacemi, přidá do své tabulky záznam o tom, od kterého zařízení tuto informaci příjemce obdržel a také informaci o tom, kolik skoků (hops) je toto zařízení vzdáleno.

V případě, že zařízení v tabulce již je, proběhne porovnání hodnoty počtu skoků v nově přijatém paketu s hodnotou v tabulce zařízení. Pokud menší je, záznam v tabulce se změní na ten s výhodnější cestou, tedy ten z nově přijatého paketu. Pokud výhodnější není, záznam zůstane nezměněn.

Takovým způsobem postupně získají všechna zařízení informaci o každém členském zařízení. V tom okamžiku už není potřeba vysílat kompletní informaci o vzdálenosti všech zařízení, informace posílaná v paketu tedy obsahuje pouze identifikátor člena a informaci, že se nic nezměnilo.

Pokud dojde k tomu, že v delším časovém intervalu zařízení nedostane informaci o některém ze svých sousedů, označí jej jako ztraceného a začne vysílat pakety informující o této

ztrátě. Tato informace se šíří skupinou, dokud některé zařízení nemá ztraceného člena jako souseda, tedy ho přímo vidí. Podobně, jako v případě ztráty zařízení, bude vysílat informaci o nově připojeném sousedovi.

Pokud ale žádné zařízení sousedem ztraceného člena není, je tento člen označen jako ztracený postupně všemi zařízeními ve skupině. Zařízení následně zobrazí upozornění uživatelům na obrazovce.

Opětné nalezení tohoto zařízení nastane v případě, že jeho vysílání zachytí zařízení některého z členů skupiny a bude o něm vysílat informaci jako o nově získaném sousedovi. Tato zpráva se poté postupně rozšíří po celé skupině i do zařízení, které původně oznámilo tuto ztrátu.

6.2 Způsob komunikace mezi zařízeními

Tato část kapitoly se zabývá návrhem základu komunikace s ohledem na požadavky na funkcionalitu aplikace a na omezení daná operačním systémem. Konkrétně se soustředí na volbu verze technologie Bluetooth a způsob předávání dat umožněný touto technologií. Dále je popsán způsob využití Bluetooth tak, aby byl v souladu s omezeními danými API poskytnutým operačním systémem Android a zároveň byla maximalizována funkcionalita aplikace, která bude tyto prvky využívat.

Volba technologie

Při řešení komunikace mezi zařízeními členů skupiny vyvstává problém volby správné technologie. Nabízí se použití Bluetooth Mesh, ta ale slouží spíše při využití v internetu věcí a navíc není v Androidu podporovaná. Bluetooth Classic se hodí spíše pro přenos velkého množství dat, například streamování hudby, navíc nabízí omezené možnosti připojování zařízení. Volba tak padá na Bluetooth Low Energy, který klade důraz na nízkou spotřebu a slouží jako nejnižší vrstva Bluetooth Mesh.

Způsob komunikace

Pro komunikaci jsem se rozhodl využít co nejvíce pakety typu advertise. Tento druh paketů přijmou všechna zařízení v okolí najednou, což je výhodnější, než zaslání stejné informace každému zařízení v okolí zvlášť pomocí navázaného spojení. Navazování spojení se ale nebude možné zcela vyhnout, je potřeba ho využít při předávání většího množství dat při změně v uspořádání skupiny.

Zařízení tedy budou vysílat advertise pakety obsahující informace o sobě, jako autorovi zprávy. Paket bude mimo jiné obsahovat data o tom v jakém stavu se zařízení nachází. Tato data slouží k předání informace zda nedošlo k žádné změně v tabulce vysílajícího zařízení a není potřeba žádné další akce, nebo jestli k nějaké změně došlo. V případě změny v tabulce je třeba získat další informace čtením nebo zápisem sdílené Bluetooth charakteristiky odesilatele zprávy. O tom jestli je tabulka předávána pomocí čtení nebo zápisu rozhodne v jakých rolích jsou spolu právě zařízení připojena.

Časování Bluetooth událostí

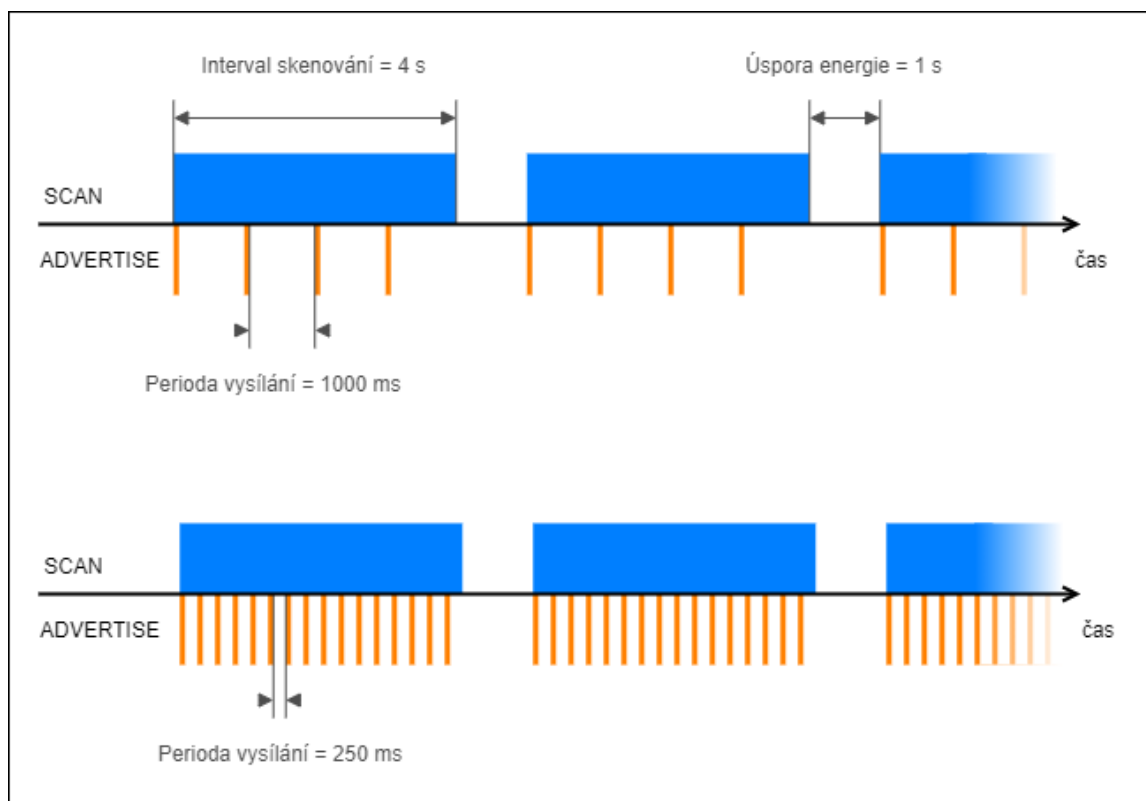
Volba časování je důležitá pro správnou komunikaci mezi jednotlivými zařízeními ve skupině.

Vzhledem k omezení počtu spuštěných skenů v Androidu na 5 v intervalu 30 sekund a tomu, že délku skenu nelze volně nastavit, jak bylo popsáno v 5.2, není možné skenovat 100 % času tak, jak to dělají uzly Bluetooth Mesh. Nejlepší podobnou volbou proto bude použít `SCAN_MODE_LOW_LATENCY`, který skenuje po dobu 4 vteřin a další 4 vteřiny je vypnutý. Pokud aplikace využije tento skenovací režim tak, že využije 4 vteřin skenování a bude jej kombinovat se sekundovou pauzou mezi spuštěnými skeny, je možné dodržet omezení dané systémem a zároveň využít maximální možný čas ke skenování. Toto omezení tak určuje interval, ve kterém bude prováděna kontrola jestli byla obdržena informace o přítomnosti všech zařízení.

Dále je potřeba zvolit frekvenci vysílání advertise paketů tak, aby se co nejvíce minimalizovala pravděpodobnost, že advertise paket nebude přijat i když jsou zařízení v dosahu. Provedl jsem proto několik měření abych určil nejlepší frekvenci se kterou vysílat advertisement.

Při použití `ADVERTISE_MODE_LOW_POWER`, který má nastavený interval na 1000 ms, se počet přijatých paketů blížil třem za dobu jedné periody skenování, tedy podle očekávání. Bohužel nastávaly i situace, kdy takový paket nebyl přijat žádný.

Zkrátil jsem proto interval vysílání na 250 ms nastavením na `ADVERTISE_MODE_BALANCED`. Při tomto nastavení docházelo k výpadkům jen velmi sporadicky.



Obrázek 6.1: Porovnání skenování a vysílání s ohledem na interval vysílání.

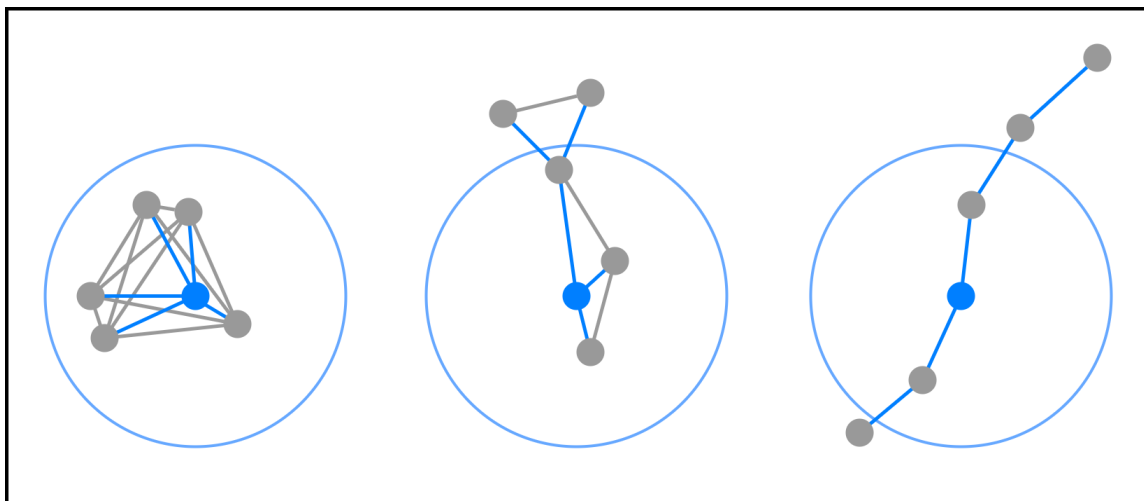
6.3 Sledování členů skupiny

Tato část kapitoly se zabývá tím, jakým způsobem bude aplikace sledovat celistvost skupiny. Nejprve se věnuje volbě topologie, potom jsou popsány pakety, kterými reagují zařízení na jednotlivé stavy, ve kterých se bude síť nacházet. Jednotlivým stavům se věnuje zbytek této části kapitoly.

Topologie

Hlavním účelem této aplikace je sledování celistvosti skupiny. Při takovém úkolu je důležitá nízká latence při předávání informací. Stejně tak je důležité brát ohled na to, že se členové budou často vůči sobě pohybovat.

S ohledem na tyto podmínky se jeví jako nejlepší řešení být připojený s co největším počtem zařízení přímo a informace zprostředkovávat až v případě, kdy je některé zařízení mimo dosah. Pro zprostředkování informace si zařízení náhodně vybere jednoho ze svých sousedů, kteří poskytují informaci o vzdáleném zařízení s nejmenším počtem skoků mezi vzdáleným zařízením a informujícím sousedem. Tak se nebude tvořit zbytečně rozsáhlá síť, když to nebude zcela nutné. Při tomto postupu vznikne síť se síťovou topologií podobně jako u Bluetooth Mesh.



Obrázek 6.2: Příklady cest, kterými modré zařízení získává informace o připojení ostatních členů skupiny. Situace jsou zobrazeny od full mesh s přímou viditelností všech zařízení (vlevo) až po situaci, kdy zařízení tvoří linii a informace o zařízeních je zprostředkovávána.

Tabulka zařízení patřících do skupiny

Aby aplikace mohla udržovat přehled o ostatních zařízeních potřebuje tyto informace mít uloženy do datové struktury. V tomto případě je to seznam objektů reprezentujících jednotlivá členská zařízení. Informace, které objekt obsahuje jsou:

- **ID zařízení** - identifikuje jednotlivá zařízení patřící do skupiny.
- **Jméno zařízení** - slouží k identifikování zařízení pro uživatele.
- **Hops** - počet skoků, mezi tímto a vzdáleným zařízením.

- **PID** - pořadové číslo paketu. Je rostoucí číslo sloužící k identifikaci paketu odeslaného z každého zařízení. Inkrementuje se na začátku každého intervalu. PID zaručuje, šíření nejčerstvější informace o každém zařízení.
- **Flag** - označuje počet po sobě jdoucích uplynutých intervalů, kdy o zařízení nepřišla žádná zpráva.
- **Device** - objekt reprezentující Bluetooth zařízení. Sloužící ke komunikaci pomocí navázaného spojení.

Pakety

Pro předávání informací ve skupině bude třeba vytvořit několik druhů paketů. Všechny pakety budou mít stejný základní formát zobrazený na obrázku 6.3.

DRUH PAKETU	ID ZAŘÍZENÍ	ID PAKETU	ZPRAVA
1 byt	1 byt	2 byty	n bytů

Obrázek 6.3: **Formát paketů.** Každý paket obsahuje informaci o svém druhu, ID odesilatele, PID - identifikátor paketu a prostor pro posílaná data. Část pro data je využita jen u paketů posílaných mezi připojenými zařízeními.

Pakety lze rozdělit do dvou skupin podle toho, ve které fázi aplikace jsou používány. První skupina paketů je využita při připojování jednotlivých zařízení ke skupině prostřednictvím poskytovatele a slouží k výměně informací potřebných při připojení zařízení ke skupině. Sekvence posílání těchto paketů je zobrazena na obrázku 6.4.

- **Advertise network**
- **Subscribe network**
- **Send ID**
- **Confirm ID**
- **Build table**

Druhá skupina paketů slouží při samotném budování a udržování skupiny. Všechny druhy paketů v této skupině ponese informaci o identitě zařízení a stavu, ve kterém se zařízení nachází, například, že nastala ztráta souseda nebo došlo ke změně tabulky. Více informací se do advertise paketu nevejde, ostatní informace jsou proto předávány pomocí navázaného spojení a čtení charakteristiky informujícího zařízení.

- **Advertise self** - tento paket zařízení vysílá pokud nedojde k žádné změně v jeho tabulce.
- **Advertise with table** - tento paket značí změnu v tabulce zařízení. Po přijetí advertise paketu zařízení přečte charakteristiku vysílajícího a zpracuje data, která přijme.
- **Info lost** - vysláním tohoto paketu zařízení oznamuje ztrátu svého souseda. Při přijetí takového paketu zařízení šíří informaci o ztrátě dál, opět pomocí vysílání Info lost paketu. K zjištění informace, které zařízení je ztracené musí příjemce přečíst charakteristiku odesilatele.

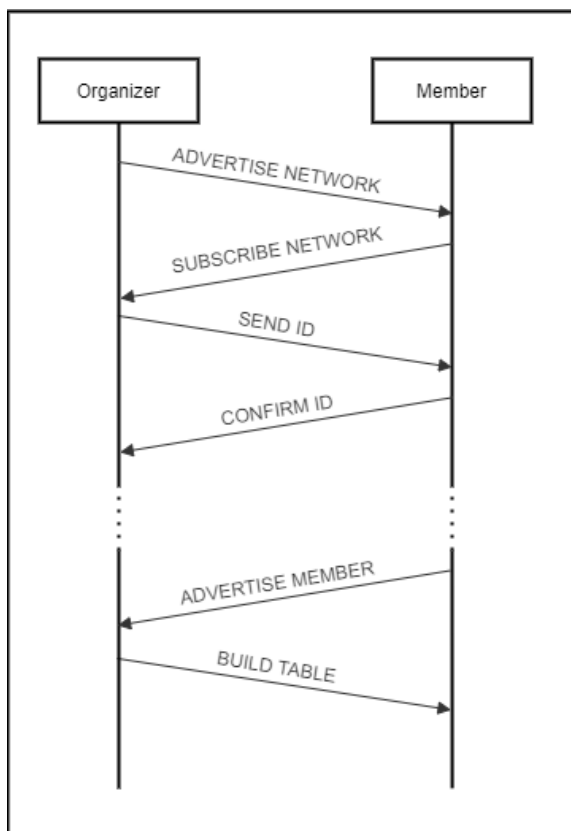
- **Info found** - paket vysílá zařízení v případě, kdy obdrží advertise paket od nového souseda nebo pokud obdrží stejný paket od zařízení, které je někým označeno jako ztracené.

Vytvoření skupiny

Ještě před tím, než bude možné sledovat celistvost skupiny, je potřeba ji vytvořit, přiřadit jednotlivým členským zařízením identifikátory a získat jejich jména. Dále je třeba určit celkový počet členů a poté tyto informace distribuovat mezi všechna zařízení. Tím bude dosažena decentralizace sítě už od jejího vzniku.

Při vytváření skupiny hraje nejdůležitější roli organizátor. Ten vysílá jméno skupiny na jehož základě se ke skupině připojují další členové. Po přijetí paketu s identifikátorem skupiny odesílá zájemce o připojení jméno, pod kterým ho uvidí ostatní členové sítě a heslo, které mu umožní se ke skupině připojit. Následně mu organizátor přiřadí ID, kterým bude členské zařízení jednoznačně identifikováno. Po přijetí identifikátoru jej zařízení potvrdí jeho odesláním zpět organizátorovi. Následně zařízení začne vysílat advertisement, informující o tom, že je připraveno přijmout paket obsahující zjednodušenou tabulku vytvořenou organizátorem, která obsahuje ID a jména všech zařízení ve skupině.

Průběh komunikace mezi organizátorem a připojovaným členem skupiny je znázorněn na obrázku 6.4.



Obrázek 6.4: Sekvenční diagram připojování zařízení (označeného jako Member) ke skupině prostřednictvím organizátora (označeného jako Organizer).

Sestavení sítě

Po tom, co organizátor předá všem členům informaci o ostatních členech právě vytvořené skupiny, začnou zařízení vysílat `ADVVERTISE_SELF` pakety a zároveň přijímají příchozí pakety od ostatních. Bez ohledu na to, jaký druh paketu zařízení obdrží, zkontroluje jestli má informaci o odesilateli ve své tabulce. Pokud ne přidá do ní informaci o tom, že je odesílatel jeho přímým sousedem - nastaví `hops` na 1, `next_hop_ID` na svou adresu a `PID` na `PID` z paketu. Pokud byl obdržený paket typu `ADVVERTISE_SELF` tím tato operace končí.

Pokud je ale přijatý paket `ADVVERTISE_WITH_TABLE`, zařízení musí ještě získat informace z odesílatelovy tabulky. To udělá tak, že přečte charakteristiku na GATT serveru odesílatele. Tato charakteristika obsahuje informace z tabulky odesílatele, konkrétně `ID`, `hops`, `next_hop_ID` a `PID` pro každé zařízení v tabulce odesílatele. Pokud je hodnota `hops` nižší, než záznam v tabulce příjemce, změní podle toho záznam ve své tabulce.

Průběžná kontrola

Když zařízení naplní tabulku záznamy o všech členech skupiny přestanou vysílat `ADVVERTISE_WITH_TABLE` pakety a vysílají pouze `ADVVERTISE_SELF`, to znamená, že v jejich tabulce nedošlo ke změně. Příjemce potom předpokládá, že zařízení reprezentovaná záznamy v tabulce obsahujícími `next_hop_ID` stejné jako je `ID` odesílatele `ADVVERTISE_SELF` paketu jsou stále přítomna.

Ztráta zařízení

Pokud zařízení neobdrží žádný paket od některého ze svých sousedů během tří 5vteřinových intervalů označí ho jako ztraceného a začne vysílat paket `INFO_LOST`. Potom očekává odpověď, že se zařízení jen přesunulo v rámci skupiny a některý ze členů je sousedem tohoto zařízení. Pokud se tak nestane v průběhu dalších tří intervalů, po celkem 5 intervalech trvajících 25 vteřin informuje uživatele o ztrátě člena skupiny.

Nalezení zařízení

Po obdržení paketu `INFO_LOST` zařízení přečte charakteristiku odesílatele a zkontroluje jestli v jeho tabulce je novější záznam o zařízení označeném jako ztracené. Pokud tam je, posílá tuto informaci v paketu `INFO_FOUND`. Tento paket opět vyžaduje aby příjemce četl nebo zapisoval charakteristiku odesílatele `advertise` paketu.

6.4 Bezpečnost

Aplikace pro komunikaci používá primárně `advertise` pakety, které může přečíst kdokoli v blízkosti zařízení, které je vysílá. Aby bylo zabráněno možnosti, že takto budou uživatelé této aplikace sledováni, jsou zvoleny dva postupy pro skrytí identifikátoru konkrétního zařízení. Prvním je náhodná volba Bluetooth adresy při zahájení každého intervalu vysílání `advertise` paketů. Druhým je šifrování paketů pomocí symetrické šifry. Symetrické šifrování využívá stejný klíč pro zašifrování i odšifrování a bude využívat heslo pro připojení ke skupině jako klíč. Hlavním účelem šifrování je skrýt identifikátor zařízení, který je součástí každého paketu, tak aby nebyl na první pohled rozpoznatelný.

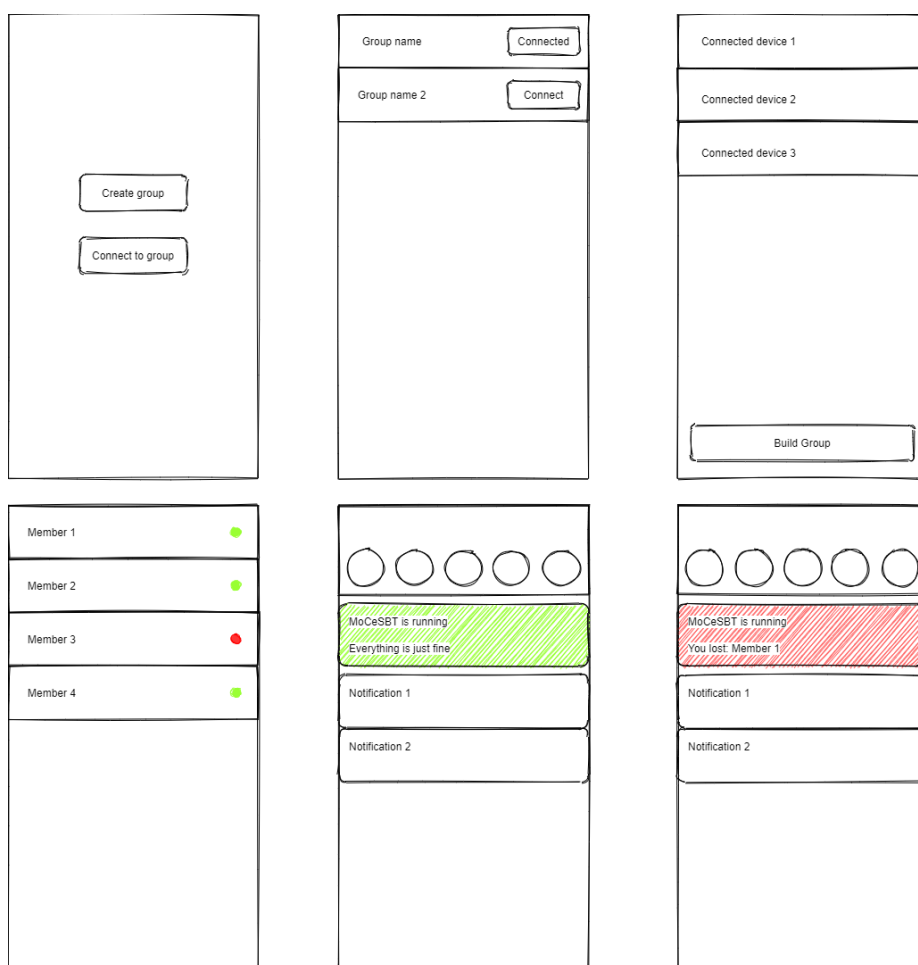
6.5 Grafické rozhraní aplikace

Uživatelské rozhraní této aplikace slouží hlavně k vytvoření skupiny a připojení jednotlivých členů. Hlavní využití - monitorování dostupnosti jednotlivých členů bude fungovat jako služba na pozadí, která v případě ztráty některého člena zobrazí ostatním notifikaci. Přesto i tato funkcionality bude mít svou obrazovku, pro kontrolu současného stavu skupiny.

Hlavními částmi uživatelského rozhraní tak budou obrazovka, kde uživatel vybere jestli bude vytvářet skupinu, nebo se připojí ke skupině, kterou vytvoří někdo jiný. Po vytvoření skupiny se všem uživatelům zobrazí seznam ostatních členů skupiny spolu s informací o jejich dostupnosti v rámci skupiny.

Důležitým prvkem je notifikace, která bude uživatele informovat o tom, že aplikace běží na pozadí a zároveň bude informovat o případné ztrátě některého z členů skupiny.

Vytvořil jsem wireframe všech těchto obrazovek, ten je zobrazen na obrázku 6.5.



Obrázek 6.5: **Wireframe uživatelského rozhraní.** Horní řada znázorňuje obrazovky sloužící k vytvoření a připojení ke skupině. Nejprve je vlevo zobrazena volba režimu, ve kterém se uživatel připojí. Potom uprostřed je obrazovka pro připojení ke skupině a vpravo je obrazovka vytvoření skupiny. Spodní řada zobrazuje část aplikace informující o stavu skupiny. Vlevo je obrazovka informující o aktuálním stavu. Vpravo jsou systémové notifikace barevně rozlišující stav skupiny.

Kapitola 7

Implementace

Tato kapitola se zabývá vlastní implementací aplikace. V první části jsou popsány nástroje použité při implementaci. Poté je popsána struktura projektu obsahující implementační detaily důležitých souborů. Konec kapitoly je věnován testování aplikace a vyhodnocení vlastností výsledné aplikace.

7.1 Použité implementační nástroje

Pro implementaci aplikace je třeba zvolit programovací jazyk spolu s vývojovým prostředím. Dále podpůrné knihovny a nástroje, které pomohou při testování a ladění.

Vývojové prostředí

Pro vývoj aplikace pro operační systém Android se jako logická volba vývojového prostředí jeví Android Studio. Toto prostředí je nabízeno Googlem zdarma, jako oficiální nástroj pro vývoj mobilních aplikací pro tuto platformu. Android Studio je postaveno na platformě IntelliJ IDEA od společnosti JetBrains. Studio nabízí podporu jazyků Java, Kotlin a C++. Dále nabízí zobrazení náhledů vytvářeného uživatelského rozhraní nebo emulátor telefonu s Androidem. Kromě těchto standardních funkcí nabízí jednoduché přizpůsobení aplikace verzi operačního systému Android. Hned při vytváření projektu Android Studio nabízí volbu nejnížší podporované verze OS Android, na které aplikace poběží. Další na co upozorňuje je využití funkcí, které nejsou v použité verzi API přístupné, podobně upozorní pokud jsou některé funkce v novějších verzích zastaralé a umožní rozdělit kód aplikace podle verze API používaného operačním systémem různých telefonů. Tím alespoň částečně usnadňuje práci na platformě s tak velkou fragmentací verzí operačního systému jako je Android.

Z programovacích jazyků, které jsou dostupné k vývoji aplikací na platformě Android, jsem zvolil jazyk Java. Především proto, že jsem se s tímto jazykem už setkal a mám s ním zkušenosti. Navíc knihovny, které jsem při vývoji využil, jsou také napsané v jazyce Java.

Knihovna Blessed

API rozhraní Bluetooth Low Energy poskytované operačním systémem Android je poněkud základní, proto se vyplatí použít některou z knihoven třetích stran, které zjednoduší práci s BLE. Mezi takové knihovny patří např. Nordic Android BLE Library, SweetBlue nebo Blessed.

Po prostudování dokumentace ke zmíněným knihovnám jsem pro programování aplikace nakonec zvolil knihovnu Blessed¹ a to především pro její jednoduchost a přehlednost. Jedná se o knihovnu vyvinutou Martijnem van Welieem v jazyce Java. Tato knihovna je volně dostupná a poskytována pod licencí MIT.

Mezi hlavní přínosy této knihovny patří:

- Poskytnutí fronty operací nad Bluetooth. Tím pádem není třeba čekat na dokončení předchozích operací.
- Je threadsafe.
- Řeší práci s vlákny, konkrétně zjednodušuje časování jednotlivých operací, zároveň řeší volbu, kterým vláknem vytvářet požadavek, a kterým vláknem následně zpracovat výsledek.
- Nadstavby sloužící k usnadnění vytváření skenovacích filtrů a párování.
- Celkové vytvoření další úrovně abstrakce zjednodušující práci s Bluetooth.

Aplikace nRF Connect

Při testování na telefonech jsem hlavně ze začátku používal aplikaci nRF Connect od Nordic Semiconductor². Tato aplikace poskytuje pro Bluetooth podobnou funkcionalitu jako Restler pro HTTP. Aplikace umožňuje uživateli skenovat okolní Bluetooth zařízení s pomocí různých filtrů a následně se k nim připojit. Je možné i zachytávat komunikaci se všemi zařízeními, připojenými buď pomocí této aplikace nebo jinak, a nabízí přímou interakci s jejich službami a charakteristikami. Dále je díky ní možné vytvořit vlastní GATT server simulující periferní zařízení, nebo lze vytvořit a následně vysílat vlastní advertise pakety.

7.2 Struktura projektu

Aplikace je tvořena dvěma hlavními částmi, těmi jsou aktivita `MainActivity` obsahující fragmenty, sloužící k interakci s uživatelem při vytváření nebo připojování skupiny, a služba `App`, která provádí komunikaci pomocí Bluetooth a má na starosti vlastní sledování členů.

MainActivity

`MainActivity` je jedinou aktivitou v aplikaci. Její instance je vytvořena při otevření aplikace. V této aktivitě se nejprve načte fragment `SelectFragment`, ve kterém uživatel zvolí zda bude skupinu vytvářet nebo se k ní připojovat. Podle této volby je spuštěn fragment `OrganizerFragment`, který slouží k vytvoření skupiny, nebo `ConnectorFragment`, pomocí kterého se uživatel připojí ke skupině vytvořené někým jiným.

OrganizerFragment

Fragment organizátora vytvoří pomocí třídy `PeripheralManager` z knihovny Blessed zařízení Bluetooth v roli `Peripheral`. To poté slouží jako server, ke kterému se připojují zařízení

¹Knihovna Blessed je dostupná z <https://github.com/weliem/blessed-android>

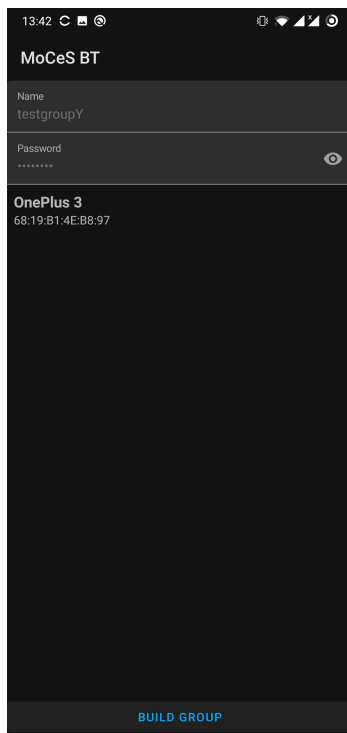
²nRF Connect je dostupné z Google Play na adrese <https://play.google.com/store/apps/details?ID=no.nordicsemi.android.mcp&hl=en&gl=US>

ostatních členů. Komunikace probíhá pomocí čtení a zápisu jedné charakteristiky, kterou organizátor sdílí.

Aby se serverem mohlo komunikovat více zařízení současně, je využit callback `onCharacteristicWrite`, který je zavolán po každém pokusu o zápis charakteristiky, ale ještě před tím, než je hodnota charakteristiky na periferním zařízení skutečně změněna. Využití tohoto callbacku umožňuje zapsanou hodnotu přečíst a nezměnit hodnotu samotné charakteristiky, která stále obsahuje inicializační informace o skupině.

Podobně jako přijímání je řešeno i odesílání. Při odesílání je využita funkce `NotifyCharacteristicChanged`. Po přijetí notifikace o změně hodnoty charakteristiky, nebo pokud se zařízení rozhodne provést čtení charakteristiky, je zavolán callback `onCharacteristicRead` a v něm je možné změnit obsah charakteristiky viditelný pro toto konkrétní zařízení.

Jakmile proběhne výměna dat mezi organizátorem skupiny a novým členem je připojené zařízení zobrazeno organizátorovi v `RecyclerView`. Uživatelské rozhraní tohoto fragmentu je zachyceno na obrázku 7.1.

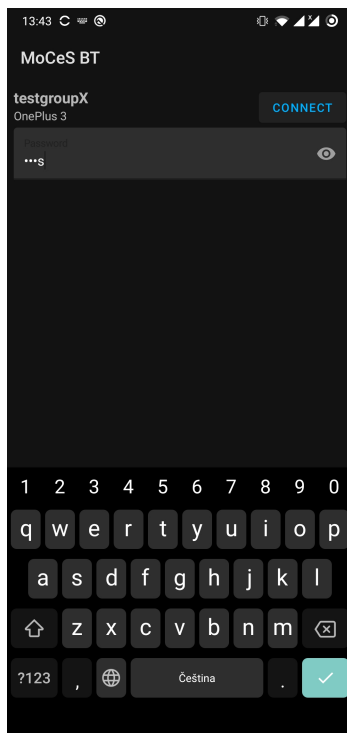


Obrázek 7.1: **Obrazovka vytváření skupiny.** Nahoře obsahuje část pro zadání jména skupiny a hesla, které jsou přístupné i po vytvoření. V dolní části se zobrazuje seznam zařízení, které se ke skupině připojily.

ConnectorFragment

Při spuštění třídy `Connector` se zařízení v rámci aplikace stává klientem nebo centrálním zařízením z pohledu BLE. Nejprve skenuje Bluetooth zařízení vysílající pakety, které obsahují UUID služby, kterou využívá tato aplikace. Po tom, co uživatel vybere skupinu, se připojí k perifernímu zařízení organizátora majícího tuto skupinu na starosti. Následně si

s ním vymění data potřebná k připojení do skupiny. Obrázek 7.2 zachycuje uživatelské rozhraní tohoto fragmentu.



Obrázek 7.2: **Obrazovka vytváření skupiny.** Obsahuje seznam připojitelných skupin.

App

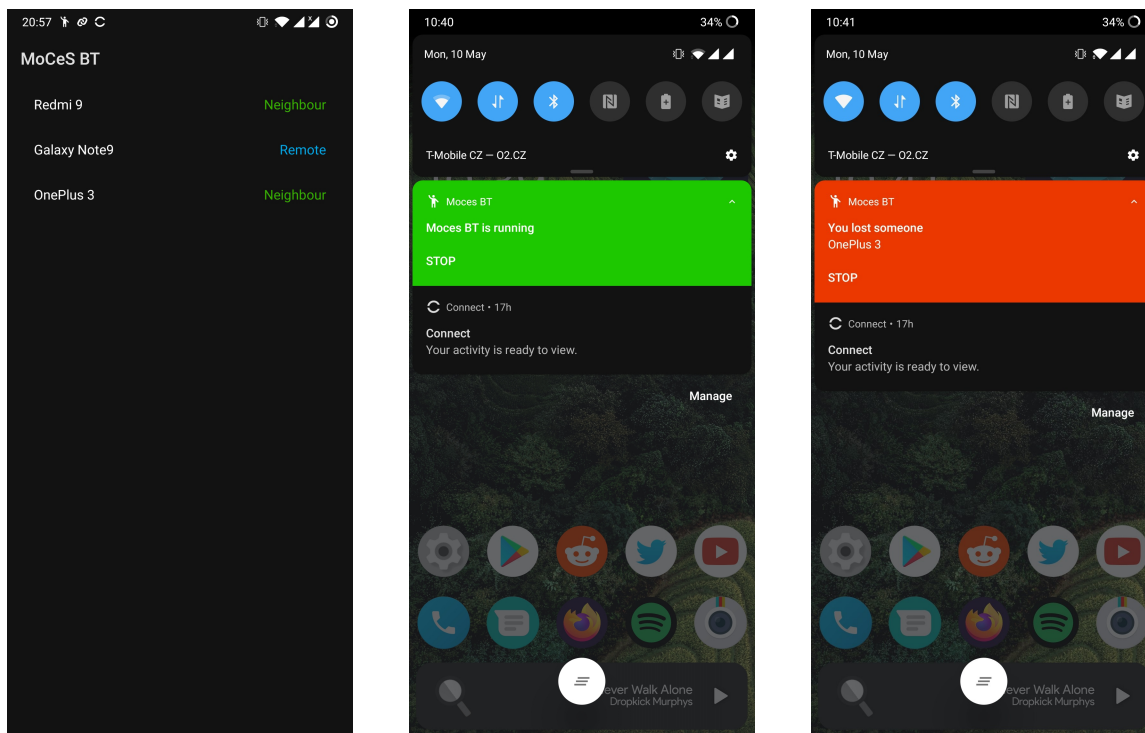
Po ukončení inicializační fáze je z `MainActivity` spuštěna služba na popředí `App`. Ta má na starost práci s Bluetooth i když je aplikace minimalizovaná nebo když má telefon vypnutou obrazovku. Kvůli předání informací o zařízeních patřících do skupiny získaných třídou `Organizer` nebo `Connector` je služba vázána pomocí `bondService`. Po spuštění služba periodicky spouští a ukončuje sken BLE zařízení v okolí, zároveň zahajuje vysílání `advertise` paketů s novým obsahem zprávy.

Při spouštění služby je vytvořena i notifikace, která je potřeba ke spuštění služby na popředí a informování uživatele, že aplikace běží. Zároveň slouží k informování uživatelů v případě ztráty člena skupiny. U zařízení s Androidem 8.0 a vyšším je potřeba spolu s notifikací vytvořit i notifikační kanál sloužící ke sjednocení notifikací z jedné aplikace. Příklady notifikací jsou na obrázku 7.3.

Za účelem zabezpečení odchozích paketů je vytvořena třída `Sec`, která paket zašifruje pomocí symetrické šifry `DESede` patřící do standardní knihovny `Cipher`. Tato šifra byla zvolena z toho důvodu, že je bloková a délka jejího bloku je 56 bitů, takže je zpráva i po zašifrování dostatečně krátká, aby se dala poslat `advertise` paketem.

DevicesFragment

Spolu se službou je vytvořen i fragment `DevicesFragment`, který slouží k zobrazení současného stavu všech zařízení ve skupině. Uživatelské rozhraní fragmentu je zachyceno na obrázku 7.3.



Obrázek 7.3: **App a DevicesFragment**. Na prvním snímku je obrazovka se zobrazením současného stavu skupiny. Druhý a třetí snímek zobrazují barevnou systémovou notifikaci služby App. První z nich zelenou barvou indikuje, že aplikace běží na pozadí a skupina je celá. Druhá notifikace je červená a informuje o ztrátě jednoho nebo několika členů.

7.3 Vyhodnocení vlastností navrženého řešení na reálných zařízeních

Aplikace byla testovaná dvěma způsoby. Protože testování různých scénářů a topologií skupiny není možné jednoduše provádět s reálnými zařízeními, vytvořil jsem pro základní testování správnosti fungování aplikace velice jednoduchý simulátor. Ten sloužil zejména k ověřování správnosti předávání informací mezi zařízeními a jejich reakcí na různé stavy skupiny.

Druhým způsobem bylo testování na zařízeních s OS Android. Tímto způsobem byla ověřována kompatibilita s různými zařízeními, komunikace pomocí Bluetooth Low Energy a spotřeba baterie. Jednou z hlavních předností Android Studio je, že poskytuje emulátor telefonu s Androidem, ten bohužel neumožňuje emulovat komunikaci pomocí Bluetooth. Veškeré testování na Androidu proto muselo probíhat přímo na fyzických telefonech.

Korektnost fungování

Testování funkčnosti aplikace probíhalo pomocí mnou vytvořeného simulátoru. Ten nahrazuje části aplikace pracující s Bluetooth přímou komunikací mezi funkcemi přijímače a vysílače. Při testu pomocí simulace byla dána vstupní topologie skupiny, která určila vztahy mezi objekty reprezentujícími jednotlivá zařízení. V průběhu simulace byly přehrávány různé scénáře a jejím výsledkem byl výpis obsahu tabulek všech zařízení v každém kroku simulace. Takto bylo testováno sestavování skupiny, krátký výpadek spojení, který

by se neměl projevit označením ztráty, dále ztráta a přesun zařízení při různých topologiích. Na základě výsledků simulace byly průběžně prováděny úpravy komunikačního protokolu.

Testování na reálných zařízeních

Reálné testování probíhalo na 5 zařízeních, které jsem měl k dispozici. Jednalo se o modely mobilních telefonů OnePlus 6, OnePlus 3, Samsung Galaxy Note 9, Samsung Galaxy A9 a Xiaomi Redmi 9. V příloze A je tabulka těchto zařízení s informacemi o verzi Androidu, verzi Bluetooth a kapacitě baterie.

Sestavení sítě

Při tomto experimentu jsem měřil časový interval od aktivace tlačítka „vytvořit skupinu“ po připojení všech zařízení a jejich označení jako připojené. Tento test jsem provedl několikrát s různým počtem zařízení. Se dvěma zařízeními trvalo sestavení 7 až 10 sekund. Při použití třech telefonů se délka sestavení skupiny protáhla na 9 až 15 sekund. Čtyřem telefonům trvalo sestavení 7 až 19 vteřin a při použití pěti zařízení trvalo sestavování 8 až 17 vteřin.

Ztráta zařízení

Nejprve jsem vyhodnocoval počet „falešných“ ztrát, kdy je zařízení sice v dosahu, ale nejsou od něj v daném intervalu přijaty žádné pakety a dojde buď k odeslání paketů `INFO_LOST` dotazující ostatní zařízení na jeho přítomnost nebo dojde dokonce k zobrazení notifikace informující o ztrátě člena skupiny. Tento test jsem prováděl postupně s několika telefony položenými vedle sebe na stole. Testy jsem prováděl 30 minut. Po dokončení jsem získal počet proběhnutých intervalů, označených ztrát zařízení spolu s počtem zobrazených notifikací.

Při prvním testu se dvěma zařízeními (OnePlus 3, OnePlus 6) byl naměřen počet označení zařízení jako ztracené 14 a byla zobrazena 1 notifikace v průběhu 225 kontrolních intervalů. Podíl špatného označení tak byl 6%.

Při druhém testu byly použity tři telefony. První dva (OnePlus 3, OnePlus 6) zaznamenaly podobný počet výpadků jako při prvním testu. V 378 intervalech došlo ke 14 a 29 výpadkům vedoucím k zobrazení 1 a 7 notifikací. To znamená chybné označení ve 4 a 7 % a notifikace 3 % a 2 %. Třetí telefon ale zaznamenal 121 ztrát a 118 notifikací. Jednalo se o model Samsung Galaxy A9. Po bližším prozkoumání tohoto zařízení jsem zjistil, že je na něm vypínán Bluetooth skener spolu s vypnutím obrazovky. Tento problém souvisí s omezením skenování při vypnuté obrazovce zavedeném v Androidu 8 popsáném v 5.2. Někteří výrobci toto omezení rozšířili na veškeré spuštěné skeny a tuto funkci není možné obejít použitím skenování s filtry jako u zařízení s neupravenou verzí OS Android. Stejný problém byl při testování zjištěn i u Galaxy Note 9. Proto jsem rozšířil aplikaci o tlačítko zabraňující vypnutí obrazovky, což má samozřejmě dopad na spotřebu baterie.

Při testování se čtyřmi a pěti zařízeními byla některá zařízení označena jako ztracená 3 a 9 % času běhu aplikace. To vedlo k zobrazení notifikací ve 2 až 3 % případů.

Dále bylo testováno měnění topologie skupiny. Tento test byl prováděn v parku čtyřmi lidmi. Skupina dynamicky měnila polohu a byl sledován stav skupiny v aplikaci. Měřením jsme zjistili, že aplikace reagovala na stav skupiny relativně přesně. Test zahrnoval vzdálení některých členů na vzdálenost mimo dosah a změnu vzájemné polohy.

Dosah

Měření dosahu proběhlo mezi dvěma zařízeními venku mimo obec a na běžné ulici. Mimo obec na prostranství s přímou viditelností byl spolehlivý dosah 40 metrů. V zástavbě, kde byly v dosahu aktivní Wi-Fi sítě a několik Bluetooth zařízení byla změřena zhruba stejná hodnota kolem 40 metrů.

Tato vzdálenost se při testování na kole ukázala jako dostatečně velká, aby nedocházelo ke zbytečným notifikacím a zároveň se člověk nemusel vracet příliš daleko v případě oddělení od zbytku skupiny.

Spotřeba baterie

K testování byla použity relativně moderní telefony vyšší a vyšší střední třídy s bateriemi o výrobní kapacitě mezi 3000 a 5000 mAh (konkrétnější informace o jednotlivých zařízeních jsou v příloze A). Při spuštění aplikace se pohybovala celková spotřeba akumulátoru telefonů mezi 4 a 5 % za hodinu, přičemž v systémovém nastavení byla spotřeba baterie aplikace určena pod 1 % za hodinu běhu. U Bluetooth Low Energy skeneru byla udávaná spotřeba 2,16 % baterie po hodině testování.

Kapitola 8

Závěr

Hlavním cílem této práce bylo navrhnout aplikaci pro mobilní telefony s operačním systémem Android, která bude sloužit k monitorování celistvosti skupiny sportovců, například při cyklistice a upozornit ostatní členy skupiny, v případě vzdálení některého z členů mimo dosah Bluetooth.

V teoretické části jsem se zaměřil na technologii Bluetooth a její varianty jako Bluetooth Low Energy a Bluetooth Mesh. Dále na součásti systému Android důležité pro vývoj této aplikace. Získané znalosti jsem následně využil při tvorbě návrhu aplikace a samotné implementaci.

Návrh aplikace spočíval ve výběru vhodné varianty Bluetooth a způsobu jakým mezi sebou budou zařízení komunikovat. Výsledná aplikace komunikuje pomocí Bluetooth Low Energy a ke komunikaci používá kombinaci všesměrového advertisementu a spojované komunikace typu central-peripheral. Aby nebylo možné jednotlivé uživatele zvenčí sledovat, je obsah advertise paketů zašifrovaný tak, aby byl čitelný jen pro zařízení patřící do stejné skupiny.

Navržený systém byl následně implementován formou aplikace pro Android a otestován. Vzniklý výsledek byl omezen volbou platformy. Operační systém Android, který je sice velmi rozšířenou platformou, totiž zavádí několik omezení při práci s Bluetooth Low Energy, především za účelem úspory baterie.

Díky této práci jsem se seznámil s principy komunikace pomocí Bluetooth Low Energy a vývojem aplikací pro operační systém Android. V budoucnu bych chtěl aplikaci rozšířit o využití GPS, díky čemuž by mohlo být označeno místo poslední komunikace se ztraceným členem skupiny. Zajímavá by také mohla být integrace nových technologií jako Bluetooth 5.1 nebo Ultra-wideband, které umožňují určení směru a vzdálenosti, ze které přichází signál od daného zařízení.

Literatura

- [1] ANFANG, H. *Bluetooth PHY – How it Works and How to Leverage it* [online], 31. prosince 2019 [cit. 2021-04-06]. Dostupné z: <https://punchthrough.com/crash-course-in-2m-bluetooth-low-energy-phy/>.
- [2] BAERT, M., ROSSEY, J., SHAHID, A. a HOEBEKE, J. The Bluetooth Mesh Standard: An Overview and Experimental Evaluation. *SENSORS*. 1. vyd. Mdpi. 2018, sv. 18, č. 8. ISSN 1424-8220. Dostupné z: <https://biblio.ugent.be/publication/8577604>.
- [3] BISDIKIAN, C. An overview of the Bluetooth wireless technology. *IEEE Communications Magazine*. 1. vyd. 2001, sv. 39, č. 12, s. 86–94. DOI: 10.1109/35.968817.
- [4] BLUETOOTH SIG, I. Bluetooth Radio Versions. *Learn about Bluetooth* [online]. 2021 [cit. 2021-04-07]. Dostupné z: <https://www.bluetooth.com/learn-about-bluetooth/radio-versions/>.
- [5] ENCYCLOPEDIA, W. the free. *Bluetooth* [online]. Wikimedia Foundation. 2021-04-14 [cit. 2021-04-15]. Dostupné z: <http://en.wikipedia.org/wiki/Bluetooth>.
- [6] GOMEZ, C. Bluetooth Low Energy Mesh Networks: A Survey. *Sensors*. 1. vyd. Basel: MDPI AG. 2017, sv. 17, č. 7, s. 1467. ISSN 14248220. Dostupné z: <http://search.proquest.com/docview/2108690652/>.
- [7] GOOGLE. *Jelly Bean* [online]. 2020 [cit. 2021-04-08]. Dostupné z: <https://developer.android.com/about/versions/jelly-bean>.
- [8] GOOGLE. *Lollipop* [online]. 2020 [cit. 2021-04-08]. Dostupné z: <https://developer.android.com/about/versions/lollipop?hl=en>.
- [9] GOOGLE. Use BLE. *Guides* [online]. 2021. revidováno 8. 3. 2021 [cit. 2021-04-01]. Dostupné z: <https://developer.android.com/guide/topics/connectivity/use-ble>.
- [10] GOOGLE. Bluetooth low energy. *Guides* [online]. 2021. revidováno 8. 3. 2021 [cit. 2021-04-01]. Dostupné z: <https://developer.android.com/guide/topics/connectivity/bluetooth-le>.
- [11] GOOGLE. Services overview. *Guides* [online]. 2021. revidováno 30. 1. 2021 [cit. 2021-04-04]. Dostupné z: <https://developer.android.com/guide/components/services>.
- [12] GOOGLE. Bluetooth low energy. *Reference* [online]. 2021. revidováno 24. 2. 2021 [cit. 2021-04-01]. Dostupné z: <https://developer.android.com/reference/android/bluetooth/le/package-summary/>.

- [13] GROUP, C. S. W. *Bluetooth Core Specification* [online]. říjen 1998, 2019-12-31 [cit. 2021-04-01]. Dostupné z: <https://www.bluetooth.com/specifications/specs/core-specification/>.
- [14] HEYDON, R., TANNER, J., ZHODZISHSKY, V. et al. *Bluetooth Mesh Models - A Technical Overview* [online]. Bluetooth SIG, Sep 2019. Dostupné z: https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=457092.
- [15] HYNČICA, O. Bezdrátové sítě typu mesh. *Automa: časopis pro automatizační techniku* [online]. 1. vyd. Teplická 88, 405 05 Děčín IX-Bynov: Automa – časopis pro automatizační techniku, s. r. o. 2005, č. 12, [cit. 2021-04-15]. ISSN 1210-9592. Dostupné z: http://automa.cz/cz/casopis-clanky/bezdratove-site-typu-mesh-2005_12_30826_1141/.
- [16] KATSANDRES, J. *Exploring Bluetooth 5 – What’s New in Advertising?* [online], 27. února 2017 [cit. 2021-04-07]. Dostupné z: <https://www.bluetooth.com/blog/exploring-bluetooth5-whats-new-in-advertising/>.
- [17] LEVIS, P., CLAUSEN, T., HUI, J. et al. *The Trickle Algorithm* [Internet Requests for Comments]. RFC 6206. RFC Editor, březen 2011. 1-12 s. Dostupné z: <https://www.rfc-editor.org/rfc/rfc6206.txt>.
- [18] LLC, A. T. *Introduction to Bluetooth Classic* [online]. Argenox Technologies LLC, 2020 [cit. 2021-04-15]. Dostupné z: <https://www.argenox.com/library/bluetooth-classic/introduction-to-bluetooth-classic/>.
- [19] SHAH, D. Gossip Algorithms. *Foundations and Trends® in Networking*. 1. vyd. 2009, sv. 3, č. 1, s. 1–125. DOI: 10.1561/13000000014. ISSN 1554-057X. Dostupné z: <http://dx.doi.org/10.1561/13000000014>.
- [20] TOWNSEND, K. Introduction to Bluetooth Low Energy. *Adafruit Learning System* [online], 20. března 2014. Revidováno 7. 4. 2021 [cit. 2021-04-07]. Dostupné z: <https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt>.
- [21] TOWNSEND, K., DAVIDSON, R., AKIBA a CÚFÍ, C. *Getting started with Bluetooth low energy*. Revised First Edition. Sebastopol, CA: O’Reilly, květen 2014. ISBN 978-1-491-94951-1.
- [22] WELIE, M. van. *Making android BLE work part 1* [online], 23. března 2019 [cit. 2021-02-23]. Dostupné z: <https://medium.com/@martijn.van.welie/making-android-ble-work-part-1-a736dcd53b02>.

Příloha A

Zařízení použítá při ověřování funkcionality

Tato tabulka obsahuje relevantní informace o zařízeních použitých při testování a ověřování funkcionality.

Model	Verze OS Android	Verze Bluetooth	Kapacita akumulátoru
OnePlus 3	9.0	4.2	3000 mAh
OnePlus 6	10.0	5.0	3300 mAh
Samsung Galaxy Note 9	10.0	5.0	4000 mAh
Samsung Galaxy A9 (2018)	10.0	5.0	3800 mAh
Xiaomi Redmi 9	10.0	5.0	5020 mAh