Brezžična in Mobilna Omrežja
Študijsko leto 2015/2016

Walkie-talkie aplikacija

Končno poročilo seminarske naloge

Rok Mirt, Jan Grilanc

Ljubljana, 5. junij 2016

Kazalo

1 Uvod in motivacija

Cilj te seminarske naloge je simulacija walkie-talkie naprav na dveh ali več pametnih telefonih, s pomočjo pripravljenega programskega ogrodja.

2 Opis uporabljene strojne in programske opreme

Za potrebe walkie-talkie komunikacije je bilo uporabljeno ogrodje, podano na učilnici. Aplikacija je bila testirana na telefonih Nexus 4 in LG G, najstarešja uporabljena verzija operacijskega sistema pa je bila Android 5.1.1.

3 Rešitev

3.1 Jedro aplikacije

Za uspešno delovanje walkie-talkie aplikacije potrebujemo vsaj dva pametna telefona, prijavljena v isto wifi omrežje. Pri tem velja poudariti, da javno dostopna omrežja niso najbolj priporočljiva - delno zaradi možnih varnostnih lukenj, v veliki meri pa tudi zaradi potencialne preobremenjenosti. Kot primer lahko navedeva kar Eduroam: ko sva testirala delovanje na omenjenem omrežju, je imelo samo ogrodje občasno precejšnje težave z iskanjem in povezovanjem s "sosednjimi" telefoni.

Glavni del aplikacije se izvaja v datoteki MainActivity.java. Tu najdemo metodo, ki ob zagonu registrira lokalni *service*, ter poskuša najti ostale naprave, vključene v wifi p2p servis.

V sami aplikaciji to izgleda tako, da je uporabnik po zagonu programa predstavljen s seznamom vseh naprav, ki so trenutno aktivne v wifi p2p omrežju. Ob kliku na neko napravo se med njo in našim telefonom nato izvede *p2p direct* povezava. Če je le-ta uspešno vzpostavljena, se odprejo vtiči (*sockets*), namenjeni prenašanju zvočnih posnetkov (ob pritisku na gumb).

Tu najdemo tudi tisti fragment kode, kjer lastnik skupine (*group owner*) sprejme povezave preko strežniškega vtiča, ter ustvari vtiče za vsakega izmed clientov. Večji del omenjene interakcije je sicer izveden v datoteki GroupSocketHandler.java.

MainActivity nič več ne poskuša izvajati dela kode, namenjenega tekstovnim pogovorom, saj slednji ne predstavlja funkcionalnosti, ki bi jo želeli ali celo pričakovali v walkie-talkie aplikaciji. V ta namen je tudi zakomentirana koda v ChatManager.java.

3.2 Obdelava zvoka

Predvajanje zvočnega posnetka, je obravnavano v datoteki VoiceManager.java.

Izvajamo ga s pomočjo AudioTrack komponente, ki ji podamo ustrezne argumente: frekvenco, konfiguracijo kanala in tip želenega audio kodiranja.

```
AudioTrack audioTrack = new AudioTrack(AudioManager.STREAM_MUSIC,
VoiceManager.FREQUENCY, CHANNEL_CONFIG, VoiceManager.ENCODING,
bufferSize, AudioTrack.MODE_STREAM);
audioTrack.play();
```

Za predvajanje prejetega glasovnega posnetka je poskrbljeno v TalkManager.java:

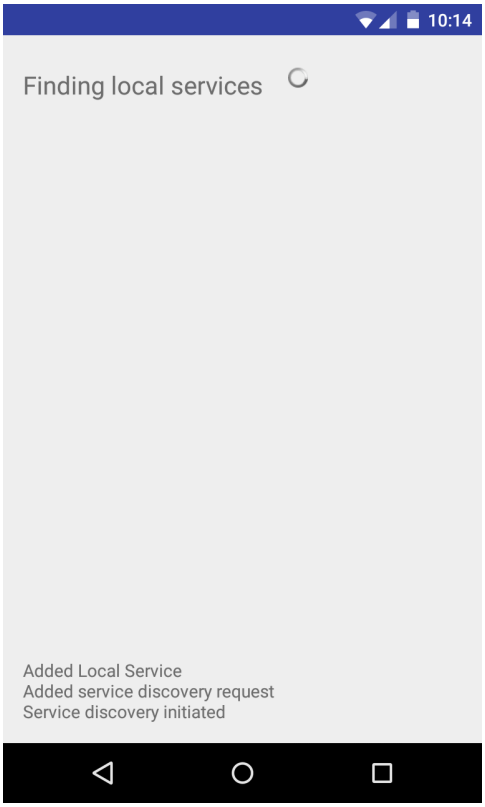
```
OutputStream out = null;
int bufferSize = AudioRecord.getMinBufferSize(VoiceManager.FREQUENCY,
CHANNEL_CONFIG, VoiceManager.ENCODING);
read = recorder.read(buffer, 0, bufferSize);
```

Na začetku sva poskušala obdelovati zvok s pomočjo komponent MediaRecord in MediaPlayer, a slednji ne znata obravnavati low level bitov, kar se je izkazalo za problematično.

4 Rezultati

Rezultat je delujoča walkie-talkie aplikacija v wifi p2p omrežju, kjer se ena naprava obnaša kot kontrolni element celotne skupine.

Takole recimo izgleda iskanje ostalih naprav, takoj po zagonu aplikacije:



Slika 1: Iskanje naprav

5 Zaključek

Prostora za nadaljne nadgradnje je seveda še ogromno. Lahko bi npr. prenovili uporabniški vmesnik; dodali zvočne signale, ki so nam bolj domači pri uporabi dejanskih walkie-talkiejev; omogočili blokiranje naprav, kar bi pripomoglo tudi k večji varnostni higieni same aplikacije.

Kljub temu so osnovni in glavni cilji aplikacije so izpolnjeni: omogočeno je iskanje naprav, njihovo povezovanje, in prenos zvoka. Obstaja torej nek konkreten temelj, na katerem lahko gradimo naprej.