

TALLINNA TEHNIKAÜLIKOOL
INFOTEHNOLOOGIA TEADUSKOND
Arvutitehnika instituut

Tarkvaraprojekt IAS1410 aruanne

Yapper

Rainer Liis 164039IACB

Silver David Iling 179620IACB

Juhendaja: Eric Anderson, Promether OÜ tegevjuht

Tallinn 2019

Autorideklaratsioon

Oleme koostanud antud töö iseseisvalt. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

Rainer Liis 164039IACB

Silver David Iling 179620IACB

Sisukord

Autorideklaratsioon	1
Sisukord	2
Ülesande püstitus	3
Ajakulud	4
Kasutatud programmid	5
Kasutajaliides	6
Programmi töö	10
Ebaõnnestumised	11
Mida õppisime	12
Kokkuvõte	13
Kasutatud materjalid	14

Ülesande püstitus

Meie projektiks oli VoIP rakendus, mis töötab kohtvõrgus. Heli kantakse üle seadmete vahel kasutades UDP protokollit. Andmepakettide suuruse vähendamiseks soovisime kasutada Opus koodekit. Kasutusele võtsime juhendaja soovil Qt raamistiku.

Ajakulud

Projekti tegemiseks kulus Raineril umbes 160 tundi, Silveril 110 tundi.

Rainer tegeles *back end*'iga - heli salvestamine, pakettide saatmine ja vastuvõtmine, heli tagasimängimine.

Silver tegeles *front end*'iga - graafiline kasutajaliides, disain.

Kasutatud programmid

Qt Creator IDE - koodi kirjutamine, projekti kompileerimine, testimine, vigade otsimine.

Qt Assistant - *offline* dokumentatsioon.

Git/Git Extensions - koodi varundamine, versioonihaldus

Audacity - testhelide muutmine.

GIMP - graafika loomine.

Wireshark - andmeside liikluse monitooring.

canva.com - projekti *mock-up*'i loomine.

vectr.com - SVG tüüpi ikoonide loomine.

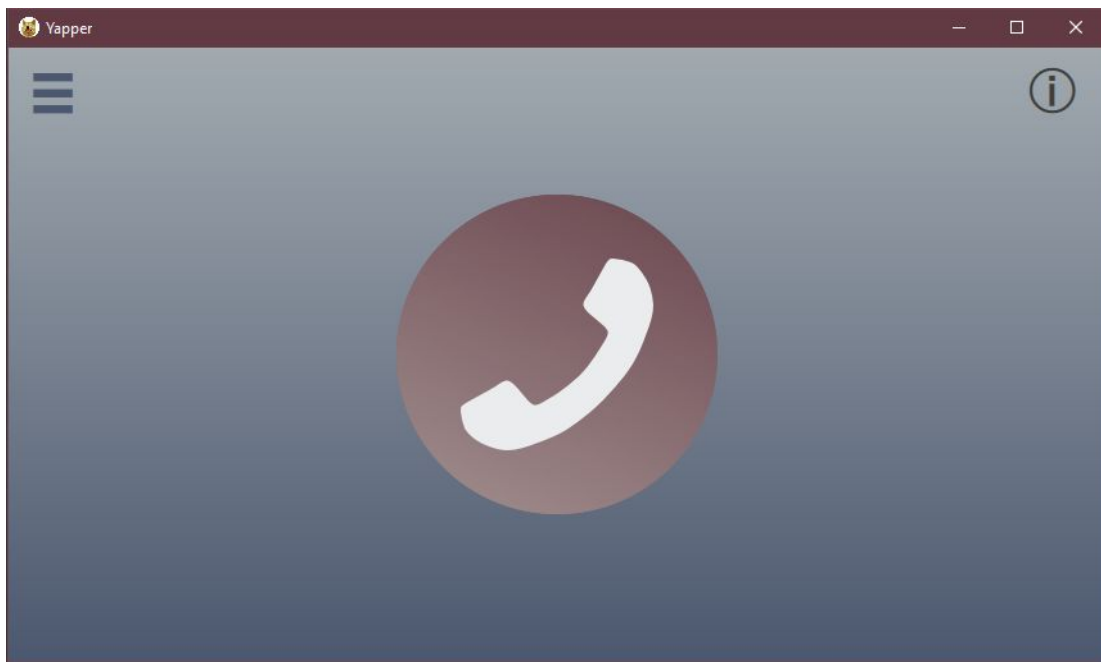
Kasutajaliides

Kasutajaliides on loodud selliselt, et rakendus oleks minimalistlik ja mugavalt kasutatav. Võrreldes näiteks teise VoIP rakenduse Mumble'iga väidan, et Yapperi kasutajaliides on kasutajasõbralikum ja näeb parem välja, olgugi, et välimus on subjektiivne. Yapperi kasutajaliides on kirjutatud puhtalt QML'iga, kasutatud ei ole Qt poolt pakutavat Designerit ega Qt Widgeteid.

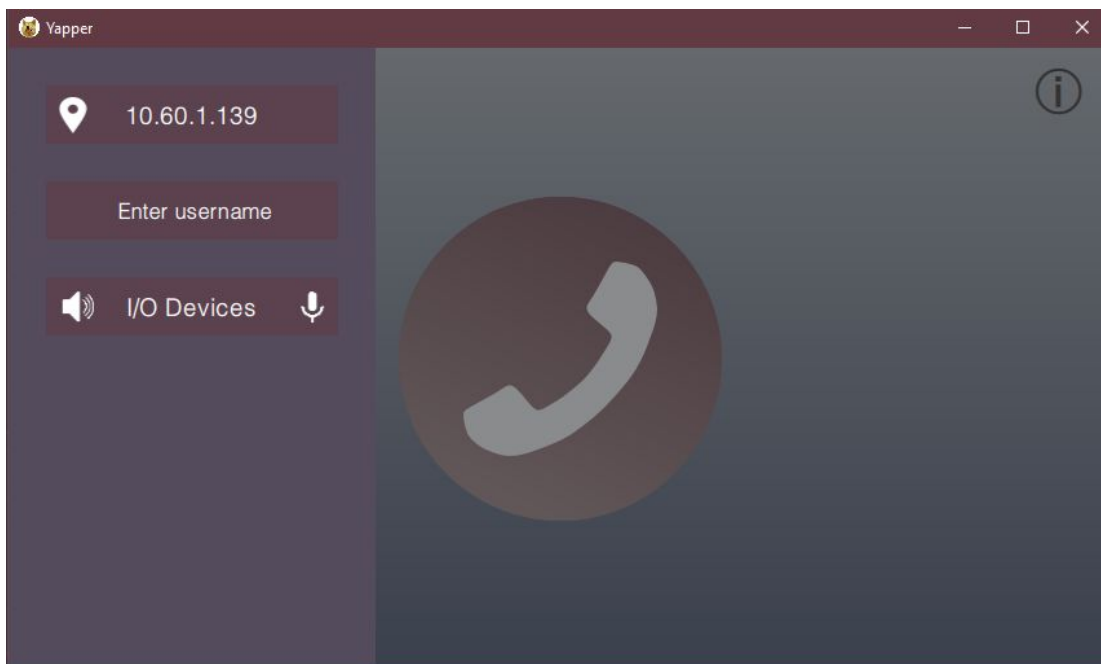
Rakendus käivitub *splash screen*'iga, mis läheb üle põhiekraanile. Põhiekraanil on kasutajal võimalus pääseda ligi menüüle, kust on näha tema enda IPv4 aadressi. Menüüst on ka võimalik liikuda sisend-väljund seadmete ekraanile, kust saab näha oma mikrofoni ja heliseadme detaile kui ka soovi korral testida mikrofoni ja heliseadme töötavust. Lisaks sellele saab menüüs muuta oma kasutajanime, mida peab igal käivitusel uuesti rakendama. Põhiekraanil on saadaval ka info rakenduse kohta, kus on kirjas, kuidas tuleb rakendust kasutada.

Helistada saab vajutades keset ekraani olevale nuppu ning sisestades IPv4 aadressi. Olles alustanud kõnet mängitakse kasutajale heli, mis kinnitab, et üritatakse ühendust luua teise kasutajaga. Lisaks sellele mängib animatsioon kuni selle ajani, mil kõne võetakse vastu, keeldutakse või möödub liiga palju aega, mis puhul ühenduse loomise katse katkestatakse ning kasutaja viiakse tagasi põhiekraanile.

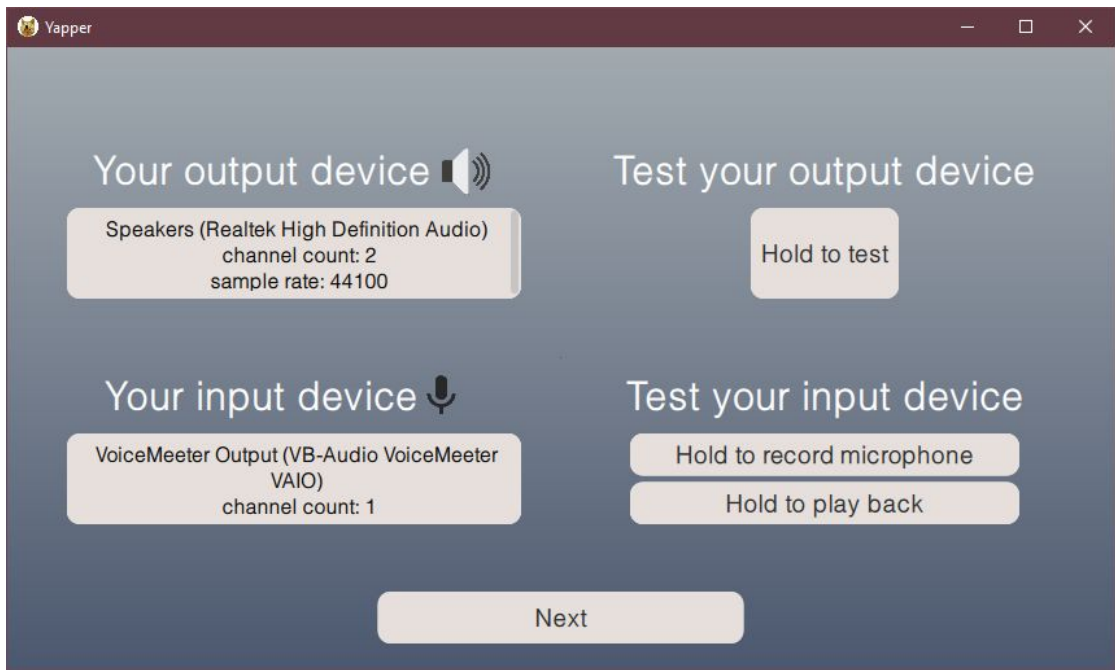
Ühenduse loomise katset alustades avaneb vastuvõtjal ekraan ja mängitakse helinat ükskõik millises vaates kasutaja ka poleks, et temale on saabumas kõne. Vastuvõtja näeb helistaja kasutajanime kui ka IP aadressi ning tal on võimalus kõne vastu võtta või kõne kinni panna. Kui vastuvõtja keeldub või piisava aja jooksul ei vasta, antakse helistajale teada, kas kõnest keelduti või möödus liiga pikk aeg. Lisaks sellele, kui vastuvõtja juhtub olema näiteks arvutist eemal ning ei tea, et temale on helistatud, siis rakendus annab kasutajale teada, et temale on tehtud vastamata kõne. Ka vastamata kõne ekraanil on näha kes, kust ja millal on temale helistatud. Kõne ajal käivitatakse mõlemal kasutajal stopper, mis mõõdab, kui kaua kõne on kestnud ja kõne ajal on ka saadaval kontroll mikrofoni ja heliseadme üle, juhuks kui keegi soovib enda mikrofoni või heliseadme vaigistada.



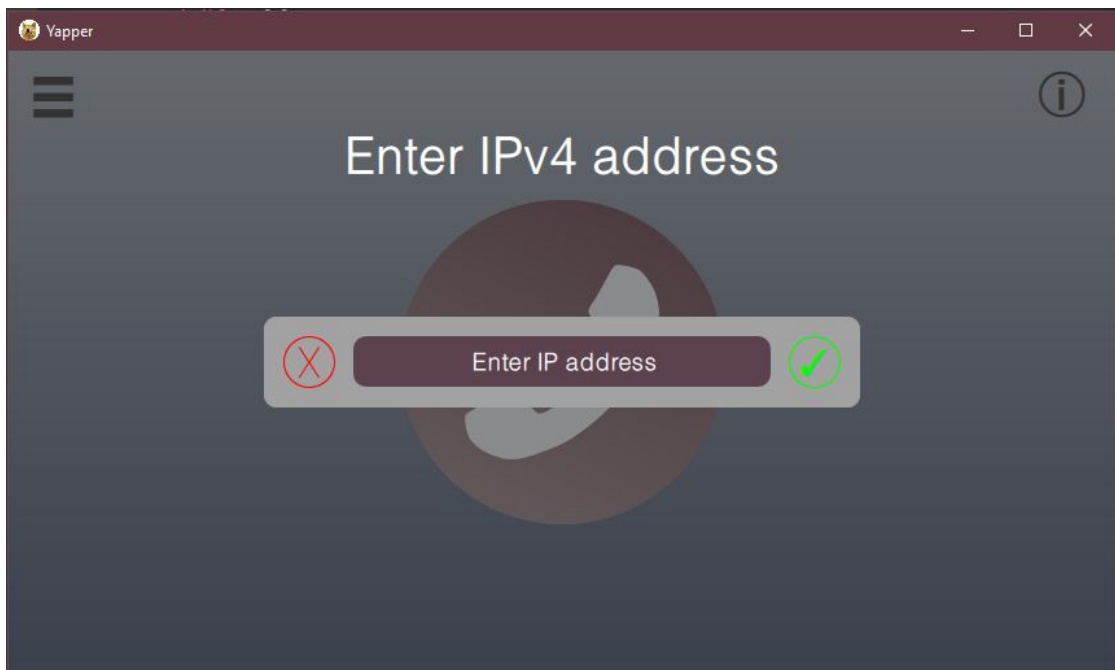
Joonis 1: Yapperi põhiekraan



Joonis 2: Menüü



Joonis 3: Sisend-väljund seadmete ekraan



Joonis 4: Kõne alustamiseekraan



Joonis 5: Kõne ootel

Programmi töö

Programmi käivitamisel kutsutakse välja funktsioonid mis panevad tööle *UDP socket*'i, väljastavad kasutajale info sisend- ja väljundseadmete kohta, kasutaja IPv4 aadressi ja määravad ära heliformaadi. Enamus C++ koodist on välja kutsutav läbi QMLi kasutades `Q_INVOKABLE` makrot.

Sisend- väljundseadmete menüüs saab testida oma seadmeid. Väljundseadmete tööd saab kontrollida mängides ette testfaili, mis on programmiga kaasas, või salvestada ise fail (mis testib mikrofoni) ja seda tagasi mängida.

Põhiekraanil menüüs saab sisestada oma kasutajanime, mis salvestatakse string tüüpi muutujasse, et seda hiljem kõne ajal kasutada saaks. Vajutades suurt nuppu põhiekraani keskel tuleb sisestada IPv4 aadress, mille järel saadetakse välja pakett, mis sisaldab märksõna kõne alustamisest ja helistaja kasutajanime. Kui helistaja ei ole määranud omale kasutajanime, kuvatakse vastuvõtjale nime "DefaultUser". Kui kõne võetakse vastu, saadetakse helistajale tagasi pakett, kus on märksõna vastuvõtmise kohta ja vastaja kasutajanimi. Kui helistaja saab kätte paketi kõne vastamise kohta, alustatakse heli ülekandmist.

Heli ülekandmine on kõige keerulisem protsess. Heli loetakse sisse sisendseadmest, salvestatakse baidijadana, saadetakse UDP paketina teisele kasutajale. Teisel kasutajal jõuab helipakett kohale, seejärel suunatakse see uude lõimesse, kus see kopeeritakse ümber ja antakse väljundseadmele, mis selle siis kasutajale mängib. See protsess toimub mõlemal kasutajal paralleelselt.

Kui kumbki kasutaja vajutab kõne lõpetamise nuppu, saadab lõpetaja välja paketi, kus on märksõna lõpetamise kohta ja mõlemal kasutajal lõppeb kõne. Sisend- ja väljundseadmed pannakse kinni ja *socket*'i algolek taastatakse.

Pakettide vastuvõtmisel on kasutatud lahendust, kus iga pakett (peale heli) sisaldab endas märksõna selle kohta, mis tegevust soovitakse ja iga paketi saamisel võrreldakse neid, et teada saada mis tegevust soovitakse. Kui ei ole teada mis paketiga tegu on, proovitakse seda helina kasutada. Kuna programmi kõik muud paketid on märgistatud märksõnadega, ei tohiks audiona võõraid pakette sisse sattuda.

Andmeside kasutus

260	86.508500	192.168.1.104	192.168.1.106	UDP	56	52104	→	4	Len=14
269	95.594438	192.168.1.106	192.168.1.104	UDP	60	4	→	4	Len=18
275	96.123146	192.168.1.106	192.168.1.104	UDP	894	4	→	4	Len=5292
279	96.141469	192.168.1.104	192.168.1.106	UDP	894	4	→	4	Len=5292
283	96.613254	192.168.1.106	192.168.1.104	UDP	894	4	→	4	Len=5292
287	96.630853	192.168.1.104	192.168.1.106	UDP	894	4	→	4	Len=5292
293	97.154695	192.168.1.106	192.168.1.104	UDP	1335	4	→	4	Len=5733
297	97.172617	192.168.1.104	192.168.1.106	UDP	1335	4	→	4	Len=5733
301	97.652768	192.168.1.106	192.168.1.104	UDP	894	4	→	4	Len=5292
305	97.660480	192.168.1.104	192.168.1.106	UDP	453	4	→	4	Len=4851
309	98.154929	192.168.1.106	192.168.1.104	UDP	894	4	→	4	Len=5292
313	98.160781	192.168.1.104	192.168.1.106	UDP	894	4	→	4	Len=5292
317	98.640594	192.168.1.106	192.168.1.104	UDP	453	4	→	4	Len=4851
321	98.647716	192.168.1.104	192.168.1.106	UDP	894	4	→	4	Len=5292
325	99.130036	192.168.1.106	192.168.1.104	UDP	894	4	→	4	Len=5292
329	99.151448	192.168.1.104	192.168.1.106	UDP	1335	4	→	4	Len=5733
333	99.615714	192.168.1.106	192.168.1.104	UDP	453	4	→	4	Len=4851
337	99.648393	192.168.1.104	192.168.1.106	UDP	894	4	→	4	Len=5292
341	100.121436	192.168.1.106	192.168.1.104	UDP	894	4	→	4	Len=5292
345	100.146626	192.168.1.104	192.168.1.106	UDP	894	4	→	4	Len=5292
354	100.621044	192.168.1.106	192.168.1.104	UDP	453	4	→	4	Len=4851
358	100.641892	192.168.1.104	192.168.1.106	UDP	894	4	→	4	Len=5292
362	101.139029	192.168.1.106	192.168.1.104	UDP	894	4	→	4	Len=5292
366	101.147335	192.168.1.104	192.168.1.106	UDP	894	4	→	4	Len=5292
370	101.640236	192.168.1.106	192.168.1.104	UDP	894	4	→	4	Len=5292
374	101.646990	192.168.1.104	192.168.1.106	UDP	894	4	→	4	Len=5292
378	102.119723	192.168.1.106	192.168.1.104	UDP	453	4	→	4	Len=4851
383	102.581582	192.168.1.104	192.168.1.106	UDP	1178	4	→	4	Len=7056
387	102.611645	192.168.1.106	192.168.1.104	UDP	894	4	→	4	Len=5292
390	102.631207	192.168.1.104	192.168.1.106	UDP	169	4	→	4	Len=3087
394	103.145900	192.168.1.106	192.168.1.104	UDP	894	4	→	4	Len=5292
398	103.153581	192.168.1.104	192.168.1.106	UDP	894	4	→	4	Len=5292
402	103.636741	192.168.1.106	192.168.1.104	UDP	894	4	→	4	Len=5292
406	103.640958	192.168.1.104	192.168.1.106	UDP	894	4	→	4	Len=5292
410	104.155187	192.168.1.106	192.168.1.104	UDP	894	4	→	4	Len=5292
414	104.169429	192.168.1.104	192.168.1.106	UDP	1335	4	→	4	Len=5733
429	104.635935	192.168.1.106	192.168.1.104	UDP	453	4	→	4	Len=4851
433	104.667514	192.168.1.104	192.168.1.106	UDP	894	4	→	4	Len=5292
440	105.137112	192.168.1.106	192.168.1.104	UDP	894	4	→	4	Len=5292
441	105.143381	192.168.1.102	255.255.255.255	DB-LSP..	248	Dropbox	LAI sync	Discovery Protocol	
442	105.144783	192.168.1.102	192.168.1.255	DB-LSP..	248	Dropbox	LAI sync	Discovery Protocol	
446	105.152254	192.168.1.104	192.168.1.106	UDP	453	4	→	4	Len=4851
450	105.616317	192.168.1.106	192.168.1.104	UDP	453	4	→	4	Len=4851
454	105.684335	192.168.1.104	192.168.1.106	UDP	453	4	→	4	Len=4851
458	106.121432	192.168.1.106	192.168.1.104	UDP	453	4	→	4	Len=4851
462	106.206349	192.168.1.104	192.168.1.106	UDP	894	4	→	4	Len=5292
466	106.621648	192.168.1.106	192.168.1.104	UDP	894	4	→	4	Len=5292
470	106.633273	192.168.1.104	192.168.1.106	UDP	894	4	→	4	Len=5292
474	107.140166	192.168.1.106	192.168.1.104	UDP	894	4	→	4	Len=5292
478	107.161690	192.168.1.104	192.168.1.106	UDP	1335	4	→	4	Len=5733
482	107.619446	192.168.1.106	192.168.1.104	UDP	453	4	→	4	Len=4851
486	107.651973	192.168.1.104	192.168.1.106	UDP	894	4	→	4	Len=5292
490	108.119067	192.168.1.106	192.168.1.104	UDP	894	4	→	4	Len=5292
494	108.148944	192.168.1.104	192.168.1.106	UDP	894	4	→	4	Len=5292
495	108.228729	192.168.1.106	255.255.255.255	UDP	82	55368	→	1947	Len=40
496	108.388141	192.168.1.106	192.168.1.104	UDP	46	4	→	4	Len=4

Joonis 6: Wireshark Yapperit kasutades ühe kõne jooksul.

440	16.542737	192.168.1.106	192.168.1.104	UDP	115	21496	→	16030	Len=73
441	16.554687	192.168.1.104	192.168.1.106	UDP	122	16030	→	21496	Len=80
442	16.563385	192.168.1.106	192.168.1.104	UDP	110	21496	→	16030	Len=68
443	16.567990	192.168.1.104	192.168.1.106	UDP	116	16030	→	21496	Len=74
444	16.582609	192.168.1.106	192.168.1.104	UDP	120	21496	→	16030	Len=78
445	16.589920	192.168.1.104	192.168.1.106	UDP	108	16030	→	21496	Len=66
446	16.603468	192.168.1.106	192.168.1.104	UDP	105	21496	→	16030	Len=63
447	16.607476	192.168.1.104	192.168.1.106	UDP	110	16030	→	21496	Len=68
448	16.622906	192.168.1.106	192.168.1.104	UDP	102	21496	→	16030	Len=60
449	16.627206	192.168.1.104	192.168.1.106	UDP	113	16030	→	21496	Len=71
450	16.642929	192.168.1.106	192.168.1.104	UDP	108	21496	→	16030	Len=66
451	16.654022	192.168.1.104	192.168.1.106	UDP	112	16030	→	21496	Len=70
452	16.663207	192.168.1.106	192.168.1.104	UDP	101	21496	→	16030	Len=59
453	16.667360	192.168.1.104	192.168.1.106	UDP	109	16030	→	21496	Len=67
454	16.682526	192.168.1.106	192.168.1.104	UDP	116	21496	→	16030	Len=74
455	16.699634	192.168.1.104	192.168.1.106	UDP	107	16030	→	21496	Len=65
456	16.701989	192.168.1.106	192.168.1.104	UDP	107	21496	→	16030	Len=65
457	16.708056	192.168.1.104	192.168.1.106	UDP	117	16030	→	21496	Len=75
458	16.722454	192.168.1.106	192.168.1.104	UDP	107	21496	→	16030	Len=65
459	16.727772	192.168.1.104	192.168.1.106	UDP	112	16030	→	21496	Len=70
460	16.741925	192.168.1.106	192.168.1.104	UDP	105	21496	→	16030	Len=63
461	16.749955	192.168.1.104	192.168.1.106	UDP	110	16030	→	21496	Len=68
462	16.762212	192.168.1.106	192.168.1.104	UDP	115	21496	→	16030	Len=73
463	16.767890	192.168.1.104	192.168.1.106	UDP	113	16030	→	21496	Len=71
464	16.782773	192.168.1.106	192.168.1.104	UDP	110	21496	→	16030	Len=68
465	16.787197	192.168.1.104	192.168.1.106	UDP	116	16030	→	21496	Len=74
466	16.801855	192.168.1.106	192.168.1.104	UDP	106	21496	→	16030	Len=64

Joonis 7: Wireshark Skype'i kasutades.

Mõõtsime mõlema rakenduse andmeside kasutust, nii et alustasime kõnet ning ütlesime sinna ühe lause ja lõpetasime kõne ära. Lause pikkus oli umbes seitse sekundit. Skype'i kasutades leidsime, et selle ajaga edastati ~1900 paketti, nende kogumaht on 228 kB.

Yapperi andmeside kasutus oli 52 paketiga umbes 47 kB.

Saadud tulemused ei peegelda aga reaalselt kasutatavust. Skype kasutab küll rohkem andmememahtu üldiselt pakette vaadates, kuid heli nendes pakettides on vähem, kui see mida Yapper edastas. Yapperil puudub koodek ning heli on katkendlik, seega antud mõõdikut ei saa kasutada reaalse võrdlusena.

Ebaõnnestumised

Programmi loomisel tuli ette ootamatuid takistusi, millega alguses ei osanud arvestada. Suurim neist on see, et meie lahenduses heli ülekandmisel peab puhvrit nullima peale iga paketi saatmist ja seega tekib heli salvestamisel iga paketi järel umbes 40ms paus, mistõttu on heli katkendlik.

Opuse koodekiga jõudsime nii kaugemale, et saime selle külge lingitud, aga realisatsioonil tekkisid uued komplikatsioonid, mille lahendamiseks enam aega polnud. Eemaldasime seega koodeki oma projektist.

Lisaks nendele tekkis probleem, et Windowsi süsteem kontrollib seadmeid ja nende muutmiseks peaksime kasutama Windowsi enda heliteeke. Kuna aega oli vähe, otsustasime et me ei lisa funktsionaalsust valida sisend- ja väljundseadmeid.

Mida õppisime

Projekti käigus õppisime palju planeerima oma tegevusi ja kohanema probleemidega.

Õppisime tundma Qt raamistikku ja QML keelt, lisaks ka C++ ja objektorienteeritust.

Õppisime kasutama versioonihaldussüsteemi Git, tänu millele oli koodi haldamine mugav ja kiire. Saime uusi teadmisi andmeedastuse kohta.

Kokkuvõte

Kuigi me ei suutnud realiseerida oma projekti täielikult, nagu soovisime, õppisime selle tegemise käigus palju ja saime väärtuslikke kogemusi tarkvaraarenduse vallas.

Link projekti lähtekoodile ja valmisprojektile:

<https://drive.google.com/drive/folders/1RCKEKODTjrFxwpLGtIKmonyVAb2MPCah?usp=sharing>

Kasutatud materjalid

- <https://www.qt.io/>
- https://opus-codec.org/docs/opus_api-1.2/index.html