TÉCNICO LISBOA

# UNIVERSIDADE DE LISBOA
# INSTITUTO SUPERIOR TÉCNICO

# Augmented communication technologies for distanced regions

**Jude Mukundane**

**Supervisor:** Doctor João Paulo Salgado Arriscado Costeira
**Co-Supervisor:** Doctor Christopher Paul Csikszentmihályi

**Thesis approved in public session to obtain the PhD Degree in Electrical and Computer Engineering**

**Jury Final Classification: Pass with Distinction**

**2022**

UNIVERSIDADE DE LISBOA

INSTITUTO SUPERIOR TÉCNICO

# Augmented communication technologies for distanced regions

**Jude Mukundane**

**Supervisor:** Doctor João Paulo Salgado Arriscado Costeira

**Co-Supervisor:** Doctor Christopher Paul Csikszentmihályi

**Thesis approved in public session to obtain the PhD Degree in**

**Electrical and Computer Engineering**

**Jury Final Classification: Pass with Distinction**

**Jury**

**President: Doctor** Mário Alexandre Teles de Figueiredo, Instituto Superior Técnico, Universidade de Lisboa

**Members of the committee:**

**Doctor** John Sören Pettersson, Karlstads University, Sweden
**Doctor** Nicola Jane Bidwell, The Technical Faculty of IT and Design, Aalborg University, Denmark
**Doctor** Christopher Paul Csikszentmihályi, Cornell University, USA
**Doctor** Teresa Maria Sá Ferreira Vazão Vasques, Instituto Superior Técnico, Universidade de Lisboa
**Doctor** Carlos Nuno da Cruz Ribeiro, Instituto Superior Técnico, Universidade de Lisboa

**Funding Institution**

Agência Regional para o Desenvolvimento da Investigação Tecnologia e Inovação

**2022**

I declare that this document is an original work of my own authorship and that it fulfils all the requirements of the Code of Conduct and Good Practices of the *Universidade de Lisboa*.

To the children, especially those of sub-Saharan Africa.

To my mother, who taught me to always believe.

To my father who resolved that even his 19th child would get a decent education.

To my sister Irene, who has always been more than a sister.

# Acknowledgement

First, I would like to give a gracious thanks to Professor Christopher Csikszentmihalyi who was more of a friend than a supervisor, and Professor Joao Paolo Costeira who took a keen interest in my work despite it having little overlap with his area of specialization. Your belief in me was fulcral to the accomplishment of this thesis.

I would like to thank the reviewers of my thesis, Professor John Soren Petersson and Professor Nicola Jane Bidwell, for their useful insights but also for their interest in my work that dates back to the early years of my research.

I would also like to thank Elbakyan for work towards increasing access to knowledge that makes it easier to do research especially in the global south.

I would like to thank different institutions that I partnered with during my research in different contexts and who provided excellent feedback, challenge and interaction with communities. These are: Northern Uganda Media Club, Bosco Uganda, University College Cork, Bere Islands Project Group, MedAlert, Active Watch, University College Cork, Association Mondiale Des Radiodiffuseurs Communautaires (AMARC).

I would also like to acknowledge the following funding institutions for their direct support as well as support to entities that I cooperated with during my research:

Acknowledgement to ARDITI - Agência Regional para o Desenvolvimento da Investigação, Tecnologia e Inovação under the scope of the Project M1420-09-5369-FSE-000001 - PhD Studentship.

Acknowledgment to ResilientAfrica Network for support under the scope of the Resilience Innovation Acceleration Project

Acknowledgement to the European Commission for support to research partner institutions under the Horizon 2020 Grassroots Wavelengths project.

# Abstract

The aim of this thesis is to develop a sustainable engineering approach for onboarding distanced communities into the digital communication space. Most practitioners have assumed that access to Internet results into increased connectivity and thereby information access and dissemination in distanced regions, whereas not. I show that more important than mere access to Internet, is mastery over the communication technologies that already fit in the economical and lifestyle constraints of distanced regions.

My research focuses on two technologies that have attained ubiquity in distanced regions – FM radio and basic telephony (voice, SMS, USSD) to concretize a platform for increased media plurality in distanced regions. The viability of this stack is largely supported by the Internet and modern digital computing without necessarily increased individual Internet access. I demonstrate an ideal situation of the Internet and modern computing to facilitate the proliferation of FM radio on minimum hardware in distanced communities.

I begin by examining the viability of low power FM broadcast in Northern Uganda to harness the ubiquity that FM broadcast enjoys over other digital technologies in distanced communities. My learnings, that minimum FM broadcast is viable but is tedious with no management tools, inform my investigation into how cloud based digital technologies can contribute to the sustenance of low-power grassroots broadcast.

The results inform the design of a minimal broadcast stack with transmission site hardware and software working in tandem with a cloud-based platform. My learnings from the deployment of this stack in two stations in rural Romania proved that minimal broadcast domains can facilitate uses and interaction models that are not possible with the current scale of FM radio. I show that both demystifying and minimalizing FM radio result in additive design of communication media as opposed to the prevailing model that proselytizes subtraction of functionality from technologies in order to deploy them in distanced regions.

I iteratively improve the technological stack that was used for the stations in Romania and observe performance in a laboratory until I arrive at a stable platform capable of sustaining broadcast over long periods. This thesis demonstrates that low-power FM broadcast on minimal hardware is a possibility; that a cloud-based platform makes this possibility viable; and that the deployment of this solution in distanced regions activates the latent information culture stifled by media singularity.

I show that the most viable way for bridging the digital divide lies not in the introduction of the Internet and modern technologies in these regions but rather using it to situate technologies in these communities in a way that allows them to use them to additively craft solutions to their communication needs.

**Key-words**

ICT4D, Community Radio, Media Plurality, FM Radio, Spectrum

# Contents

# List of figures

# List of tables

# List of acronyms

| | |
|---|---|
| AAC | Advanced Audio Coding |
| ALSA | Advanced Linux Sound Architecture |
| AMQP | Advanced Message Queuing Protocol |
| AWS | Amazon Web Services |
| AI | Artificial Intelligence |
| CAPEX | Capital Expenditure |
| CBO | Community Based Organization |
| CE | Conformité Europeenne |
| DTMF | Dual Tone Multi Frequency |
| ERP | Effective Radiated Power |
| XLR | External Line Return |
| FM | Frequency Modulation |
| GPIO | General Purpose Input/Output |
| GPS | Global Positioning System |
| GSM | Global System for Mobile |
| GOIP | GSM Over IP |
| HAT | Hardware Attached on Top |
| *ICTD* | Information and Communication Technology for Development |
| IVR | Interactive Voice Response |
| IDP | Internally Displaced Persons |
| ISP | Internet Service Provider |
| ITM | Irregular Terrain Model |
| ITWOM | Irregular Terrain Model With Obstruction Modelling |
| JAR | Java Archive |
| JMF | Java Media Foundation |
| JSE | Java Standard Edition |
| JVM | Java Virtual Machine |

| | |
|---|---|
| KYC | Know Your Customer |
| LTE | Long Term Evolution |
| LRA | Lord’s Resistance Army |
| MPX | Multiplexed Signal |
| NAT | Network Address Translation |
| OLPC | One Laptop Per Child |
| OSHI | Operating System and Hardware Info |
| OPEX | Operational Expenditure |
| OEM | Original Equipment Manufacturer |
| OTT | Over The Top |
| PNG | Portable Network Graphic |
| PPM | Portable Pixmap |
| PSTN | Public Switched Telecommunication Network |
| PCM | Pulse Code Modulation |
| PCMA | Pulse Code Modulation - A Law |
| PCMU | Pulse Code Modulation - U Law |
| RU | Rack Unit |
| RF | Radio Frequency |
| RTC | Real Time Clock |
| RSS | Really Simple Syndication |
| RTP | Real-time Transport Protocol |
| RSSI | Received Signal Strength Indicator |
| REST | Representational State Transfer |
| RTCP | RTP Control Protocol |
| SCP | Secure Copy Protocol |
| SSH | Secure Shell |
| SIP | Session Initiation Protocol |
| SRTM | Shuttle Radar Topography Mission |
| SWR | Standing Wave Ratio |
| TTS | Text To Speech |

| | |
|---|---|
| USSD | Unstructured Supplementary Service Data |
| VNC | Virtual Network Computing |
| VOIP | Voice Over IP |
| WAV | Wave Audio Format |

# Chapter 1

## Introduction

This chapter lays out the motivation behind, and key hypotheses formulated for, my research. I specify the main contributions of my research and the chronology of my research activities

## 1.1 Introduction

The conception of contemporary communication technologies is significantly premised on the ubiquity of broadband Internet. An examination of trends, however, reveals that for the foreseeable future communities that are socially, economically, politically, and geographically distant from the centers of production of technology will still face challenges in accessing – let alone using – the Internet. Attempts have been made to introduce modern communication technologies in distanced communities, but research shows that while much effort has gone into increasing access, usability is compromised partly because there has been scant opportunity for these communities to influence the design of the technologies that they will eventually consume.

This thesis proposes a design in which modern digital ICTs foster ownership and design of FM radio in the distanced communities which depend on it the most, and subsequently transform members of these communities into users of the Internet and its associated modern computing constructs. As I will show, the Internet and modern computing allow for a new FM radio model that is sustainable at the lowest echelons of the socioeconomic scale, while simultaneously addressing long-held regulatory concerns. In return, communities depending on these stations make demands of the technology that powers them, influencing design in the digital space even while demonstrating low levels of individual usage of the Internet.

My thesis is a series of pioneering efforts to shrink FM radio to a scale in which its ownership and operation are within the capabilities of distanced regions. This gain in the analog sphere is afforded by advances in digital computing and emergent societal constructs that have been observed in the communities where I conducted my research. By accessing and owning a medium that they both rely on and understand well, distanced communities can contemplate it and make decisions about its evolution with regard to their interests. These decisions have a direct bearing on the digital technologies facilitating this scale of FM radio in the first place. I arrive at what is arguably the most established idea regarding the Internet of FM radio.

In developed regions, the Internet has grown to subsume previously ubiquitous communication media, establishing itself as the de facto means of connectivity. In deviating from previous practice that produced communication technologies such as analog FM radio and Television, which were specialized in function, the Internet, by contrast, is designed for no single communication model, but effectively for any conceivable electronic communication need. While the Internet is ubiquitous in most developed regions, its proliferation and use in socially, economically, politically, and geographically distanced regions remains a more significant challenge than most people – even informed technologists – are aware of.

This thesis highlights the potential to increase and scaffold communication networks in regions where the Internet is either unavailable or a barely usable tool in the prevailing electronic communication media setup. I show that the devolution of control over FM radio, coupled with appropriately sized audiences, results in relaxed constraints and novel, organic uses that better suit the idiosyncrasies of communities distanced from the mainstream and targeted by the designers of modern digital communication technologies. In addition, I arrive at a technological solution that addresses the concerns concomitant with the devolution of control over the FM spectrum and the proliferation of low-power FM radio stations.

I bring to the fore variables that result in some technologies working exceptionally well in some communities, whilst being unusable in others. I clearly identify the limits of the Internet and subsequently segue into technologies designed to surpass these limits. The overarching goal of my thesis is to bootstrap an electronic information culture within distanced communities, as an initial step to increasing the participation of these communities in the digital space. I demonstrate that communities can use their low-power FM radio stations as a nucleus for a communication platform that incorporates components of modern digital platforms.

I put an emphasis on ownership and control because communication technologies often end up in regions far from those for whom they were designed, in many cases propelled by the utility enjoyed by their target consumers. The control of these technologies, however, is in many cases removed from fringe communities, as many of the communication technology providers in developing and distanced regions are devoid of these communities. Without the ability to contribute to the conception of these technologies, distanced and/or developing regions can only influence technological trends by innovating within the economic, social, and political constraints built into the technologies that they make use of.

I work principally with two technologies that have been broadly adopted, even in the most restricted conditions: FM radio and basic cellular telephony (voice telephony and SMS). FM radio is widely available, but its control is centralized, and typically it is used to broadcast messages from atop the socio-economic and political echelons to grassroots communities, leaving communities that depend on it with no viable options to use it for intra-community communication. Basic cellular telephony, on the other hand, while accessible to most populations, is barely used, with some of the largest markets recording paltry minutes of use per month per capita.

In addition, I utilize speech and audio technologies to facilitate the creation and dissemination of information within grassroots communities, but also as a bridge between the Internet, which is mostly written, and distanced regions in which orality is the principal mode of information exchange. Finally, I present pioneering work that aims toward a technological suite for ultra-low-power FM radio with consideration for concerns raised by communication regulators. My research yields the implementation of a technological stack for viable, sustainable, and appropriately sized community FM radio, and subsequently a model that can be applied to other community initiatives in the telecom and ISP spaces.

## 1.2 Dissertation statement

The constraints and requirements of modern communication technologies leave FM radio and mobile telephony as the two most promising technologies that can be optimized for electronic communication within distanced contexts. The prevailing modes of deployment and regulatory regimes allow for these technologies in distanced contexts, but do not necessarily facilitate their ownership and use.

The Internet and modern digital computing can facilitate appropriate scaling of these technologies and hand their control over to grassroots communities in a manner that both promotes innovation, stemming from increased sustainable use by said communities, and complies with existing regulatory regimes.

## 1.3 Dissertation overview

I begin my dissertation with a definition of what I consider distanced regions to be, and give the reader a brief introduction to the state and trends of connectivity in distanced regions in chapter 1. In chapter 2, I discuss works geared towards bridging the information divide and that share the same overarching goal as my thesis. I briefly explore their achievements, limitations, and the space they leave for RootIO to fill. In chapter 3, I examine in detail the motivation behind, and implementation of, my intervention, RootIO, around which this thesis is centered.

In Chapter 4, I discuss the findings from prototyping the RootIO idea with two communities in Northern Uganda, with reference to data from surveys and interviews, which provide insights into both the lifestyles in these communities and their interaction with communication technologies. The lessons learned from deployments in Northern Uganda shaped the first RootIO hardware and software stacks, which were deployed in other contexts in Europe and West Africa. In chapter 5, I detail the use of the RootIO platform in two communities in rural Romania and, where relevant, refer to lessons from concurrent deployments in Portugal and Ireland.

In chapter 6, technical lessons from deployments in Romania inform the final iteration of the RootIO platform, and I present key indicators regarding the performance of the RootIO stack in a lab setting. The technical results are promising, and the social indicators affirm that RootIO is capable of enabling grassroots FM radio and bootstrapping an electronic information culture in distanced communities where individual Internet use is far from tenable. Chapter 7 lays out what I term “assumptions;” that is, factors that were tractable for myself and the partners I worked with on the RootIO project, but that may prove challenging for communities seeking to replicate FM radio stations. I conclude by pointing to future work that builds on, and strengthens, the achievements of the RootIO initiative, most of which is already in progress.

## 1.4 Motivation

My research is chiefly influenced by my experiences: from using GSM feature phones and FM radio in rural Uganda, to becoming a software engineer for communication solutions in a telecommunications company, and finally as a user and researcher of sophisticated Internet-driven communication technologies in Western Europe. I have lived on either side of the digital divide and been involved in efforts that seek to bridge or narrow this divide. I have been on the pushing side: innovating to drive communication solutions deeper into distanced regions, as well as on the pulling side: devising ways to access and use modern communication technologies in developing regions.

I begin by looking at how communities interact with FM radio and mobile telephony within the constraints imposed by economic and regulatory regimes. I then connect the work of engineers and designers in the digital space, some of whom have never owned an FM radio, with populations that have, in some cases, never used the Internet. By realizing a stack that links analog FM radio with digital technologies, I bring the concerns of distanced communities into a space that is currently receiving increased attention and give FM radio, and the communities that depend on it, access to the advantages afforded by digital computing. In the background, my thesis assembles cloud computing, speech technologies, IP telephony and AI to bring automation into the space of FM radio, relieving it of long-held obligations that limited its ownership to entities at the top of the socio-economic scale.

The overarching goal of my research is the facilitation of ownership and control over communication media at the grassroots level. I argue that this is a better path toward plurality in future communication media. I also demonstrate that if the entities supplying communication technologies in distanced regions are appropriately sized and governed in the communities that they serve, then one can relax the constraints which are currently intended to safeguard the use of said technologies in contexts far removed from their centers of origin and control. I draw parallels with examples such as the relaxation of “Know-Your-Customer” (KYC) requirements in the banking sector in rural Indian contexts in a bid to increase financial inclusion [41]. The results from focusing on particular communities bring to the fore novel design considerations that would otherwise be overshadowed in generalized solutions.

My research paves the way for the additive construction of community radio, in which every component is evaluated before it is included in the setup of a particular community. Prevailing community radio is mostly subtractive, it is arrived at by removing technical and regulatory components from commercial and national broadcast. The result of this is that community radio is rarely seen as its own medium, but rather an inferior version of commercial radio.

I formulate a pragmatic approach in which I avoid building technologies from scratch, because to do so would usher in a host of distribution and logistical commitments, but rather examine novel assemblies of existing technologies and use the lessons learnt to make proposals regarding the future designs of communication media. A return to the drawing board would present little incentive for communities already trying to find solutions for numerous challenges and to do so is mostly the prerogative of communities with an abundance of academic and economic resources to invest in the process; a stark contradiction to the prevailing conditions in distanced regions.

### 1.4.1 Defining distanced communities

In my thesis I focus on communities that I term as distanced. At first mention, this often brings to mind geographically distanced rural and remote areas in which service delivery is below the standard offered in the metropolis. Physical distance is a direct challenge to the deployment of communication technologies: poor road networks hinder installation and maintenance operations, electricity supply is in many cases not readily available or is unreliable, as I witnessed in rural Romania and Northern Uganda during my research, and populations are scattered resulting in higher cost per served user.

The decrease in service delivery means that populations living in **geographically** distant regions lack opportunities in key areas such as education and healthcare, factors that are crucial to earning higher incomes; as such, an **economic** distance emerges, facilitated largely, but not entirely, by geographic distance. Economic distances persist even in urban centers, but in developed regions where the middle class is the majority [63] and populations are concentrated in urban centers, this distance becomes less apparent. In the minds of designers in developed regions, targeting the urban middle-income class is as good as designing for the masses, but their technologies end up in communities where middle classes are barely existent, and populations are mostly rural. This means, in turn, that a technology in a rural community will see a significantly different pattern of usage, relative costs, and functionality than it would in a more urban context as illustrated by Wyche and Murphy [119].

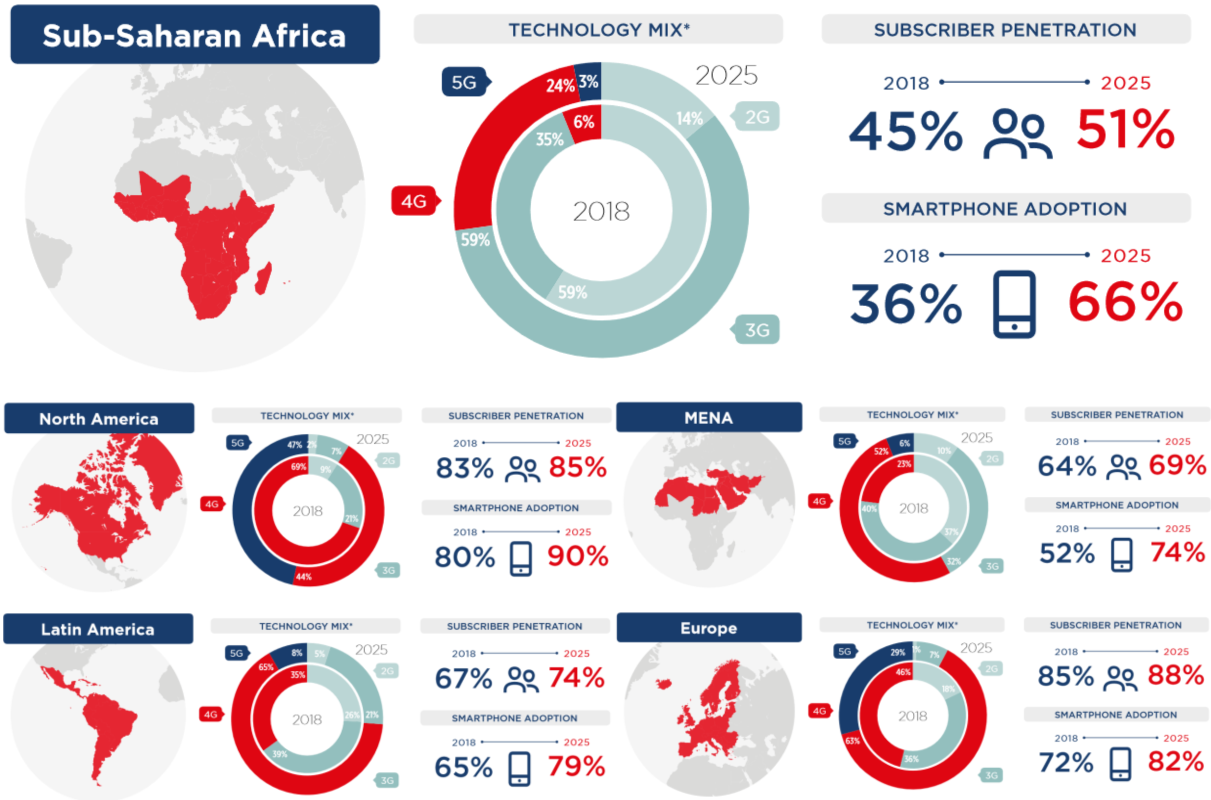
Another dimension of distance that I consider but that is often overlooked in technological discussions is **political** distance, which is often related to racial or ethnic identity. This emerges when certain communities find themselves far-removed from the centers of power responsible for formulating policies governing the larger society to which they belong. Elazar [33] detailed the nature of political distance in the United States of America and the corrective steps taken to bridge this distance. Unfortunately, while this distance still manifests itself in various communities, little has been done to address it in many areas. From the Rohingya of Myanmar [72] to the Uighurs in China [44], to the millions of refugee populations living in Eastern Africa [111], these populations are often unable to contribute to the laws governing them and are forced to abdicate their freedoms.

One last element of this distance to consider is that of **social** distance, which is the distance between individuals belonging to two distinct groups. In ethnically diverse countries this often forms along ethnic lines, while in ethnically homogenous countries this may form along economic or religious lines. Borgadus' scale for evaluation of social distance [13] evaluates the sentiments that one person or group may feel towards a category of persons. Social distance is potentially significant for political and economic distances if some denominations are ostracized. Social distance is the hardest to address given its causes: while geographic distance may be solved by moving, economic distance by uniform earnings and political distance by increased representation, it is hard to bridge the gaps in social class, race and ethnicity, gender, or sexuality.

A distanced community in my thesis is therefore defined as a community that exhibits, in at least one of these dimensions, **a substantial difference from the demographic mainstream targeted by designers of digital technologies**. While in some cases, aspects of these four types of distance may be factored into a technology's design, this thesis seeks to propose a solution that addresses the constraints posed by social, economic, political, and geographic distance from the general demographic of the countries in which I carried out my research.

### 1.4.2 State of connectivity

A perusal of literature about Internet connectivity by the GSM Association (GSMA) [45] reveals an evolution of the thinking about what it really means to be connected. In earlier literature, mobile penetration was growing, estimated at 67% as of 2019. In the same year in Europe and North America, subscriber penetration was upwards of 80%, while in sub-Saharan Africa penetration was at a paltry 45% and growing by a declining rate of under 1% per year.



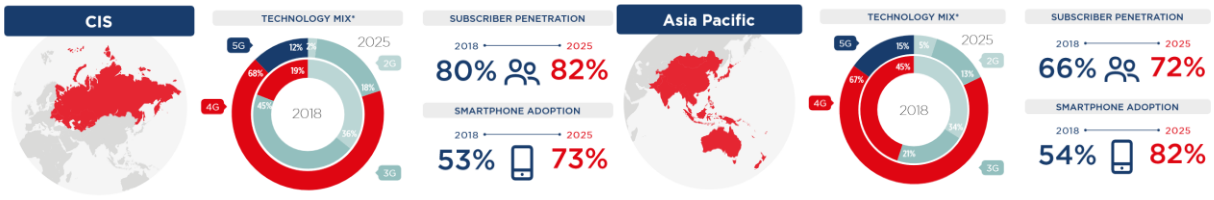



**Figure 1.1: Infographic showing global mobile subscription and smartphone adoption**

Mobile phone penetration, on the other hand, was touted as growing globally, but at a higher rate in in developed regions than in developing regions. The statistics in developing regions were further diluted by multiple subscriptions by the few who are connected [104]. Developing regions also exhibit the least smartphone penetration, at an estimated 36% of all mobile phones owned [45]. In designs that require the Internet as the principle transport medium, smartphones matter: Ericsson speculates that by 2023 smartphones will account for 95% of broadband subscriptions, and a similar share of mobile Internet traffic [123]. In sub-Saharan Africa the GSMA predicted a smartphone penetration of 33.6% by the end of 2025, the lowest globally.

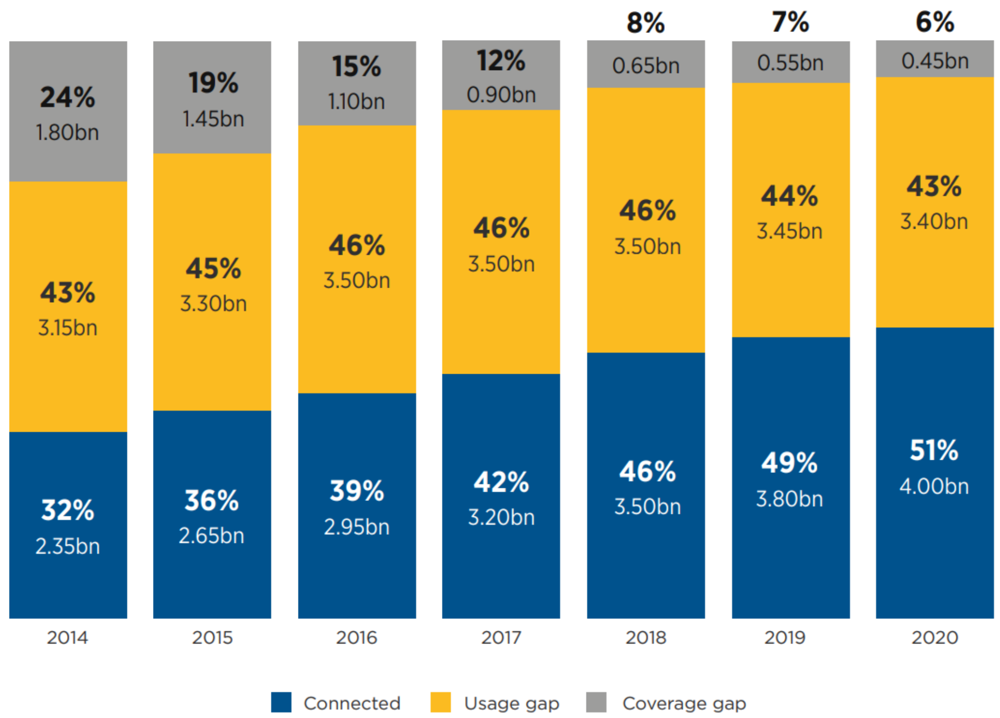



**Figure 1.2: Graphic showing global population trends in relation to coverage, connectivity and use of the Internet**

More recent statistics from the GSMA however bring more factors to the table – most notably that coverage, access and usage are three very different dimensions of the connectivity problem. The coverage issue has been addressed by increased rollout of telecommunication networks resulting in a reduction of the coverage deficit from 24% in 2014 to 6% in 2020. Of the estimated 450 million people without coverage, 210 million are resident in sub-Saharan Africa.

Recent statistics from the GSMA feature a new dimension – the usage gap, defined as the number of people that are covered by internet-capable networks but are not able to use the Internet [28]. The reduction in the coverage gap has actually resulted in an increase in the usage gap in Africa and South Asia, meaning that the Internet in its current form is for some reasons still not usable in these populations even though it is accessible.

With coverage fast becoming global, the discussion brings to the fore the speed of available Internet-capable networks and the devices used to access them. The mobile scene in Africa is still characterized mostly by feature phones and 3G capable devices, limiting the use of modern features and rich media that characterize the prevailing Internet. It can be said that access to faster communication technologies will be untenable for a substantial population in developing regions due to the scarcity of necessary devices.

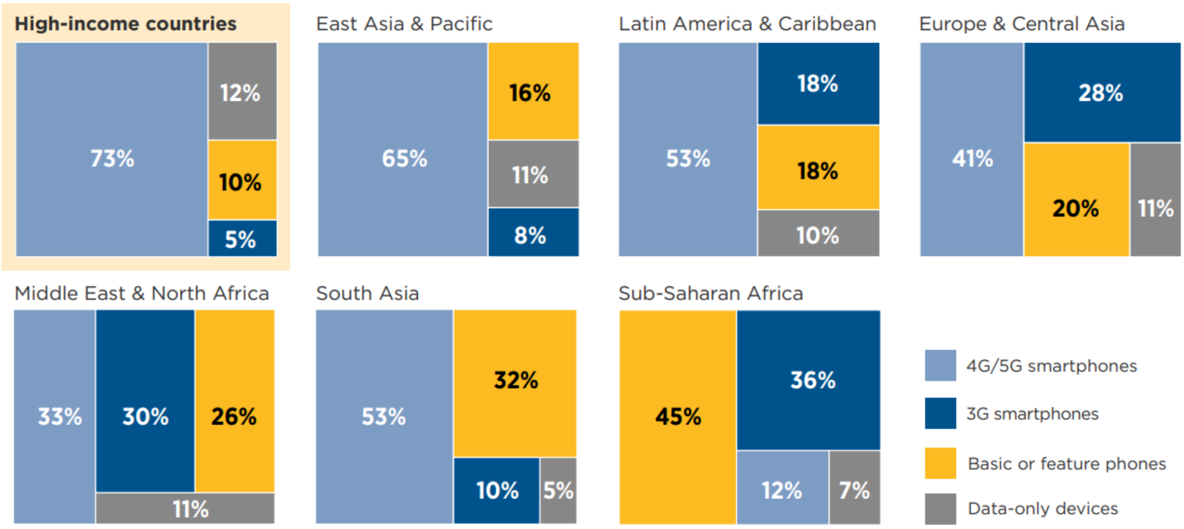



**Figure 1.3: Graphic showing the composition of mobile devices across different geographic regions**

The Internet is replacing previously ubiquitous communication technologies not only in use but also on the RF spectrum: the deployment of 5G, for example, appropriates the spectrum in three major bands, including the second digital dividend used for UHF analog television [127]. In communities where the Internet is relatively less ubiquitous, a shift to 5G would require the decommissioning of a more commonly used technology such as analog television. This would have stark consequences for communities that have not figured out how to use the Internet. Below I list some of the persistent issues that rationalize a design that relies less on individual Internet use than electronic communication media that already enjoys ubiquity in developing regions.

Delving into individual economies reveals communities with worse demographics than regional averages. For example, a report from the GSMA [124] shows that rural areas are less covered than urban areas, implying worse access than the already grim regional and national averages. It is also revealed that there is a chasm in access when it comes to genders. In Kenya for example it is shown that smartphone ownership in women has grown by only 2% to 34% between 2019 and 2022, compared to a 7% growth to 49% in men for the same period [128]. This, combined with other factors such as lower literacy in women eventually results in a 37% and 41% gender gap in mobile internet use in sub-Saharan Africa and South Asia respectively [128].

## 1.4.3 Challenges to connectivity in distanced contexts

The challenges to the deployment of the Internet in developing contexts may be obvious, but the factors that hinder the use of these technologies once deployed are often the subject of assumptions or are not considered by the designers of modern communication technologies. I will shortly discuss these challenges and distinguish between those that can be overcome, to deploy technologies in the most challenging contexts, and those that affect the subsequent use of these technologies.

The sheer availability of communication technologies does not necessarily result in their use. In the communities in which I conducted a study of the deployment of RootIO FM radio stations I observed, for example, that most people who relied on FM radio for information access had never used it to send or solicit information (see survey results from Pabbo town in chapter 4.1.2). In relation to this, while mobile phones are the prominent electronic medium for two-way communication, their use is minimal in developing regions – even lower than in regions where users have access to other electronic media for interaction [28].

The trends below imply that, for the foreseeable future, a considerable stratum of the human population will be unable to benefit from the current design of Internet-based communication technologies. My thesis delivers a solution that does not dismiss these factors, but rather considers them as representative of the norm in many developing regions.

### 1.4.3.1 Challenges to connectivity: Infrastructure paucity

Faster communication technologies implicitly require increased infrastructural development. Increasing generations of mobile telecommunication technologies mandate ever decreasing cell sizes and are more suited to urban areas with dense populations, established energy grids and transport networks, in contrast to the norms in developing regions such as sub-Saharan Africa, or even the rural Romanian communities that I visited in the Danube delta. Calderón and Servén [17] empirically showed increased growth and reduced inequality resulting from higher quantity and quality of infrastructure in Latin America and the Caribbean.

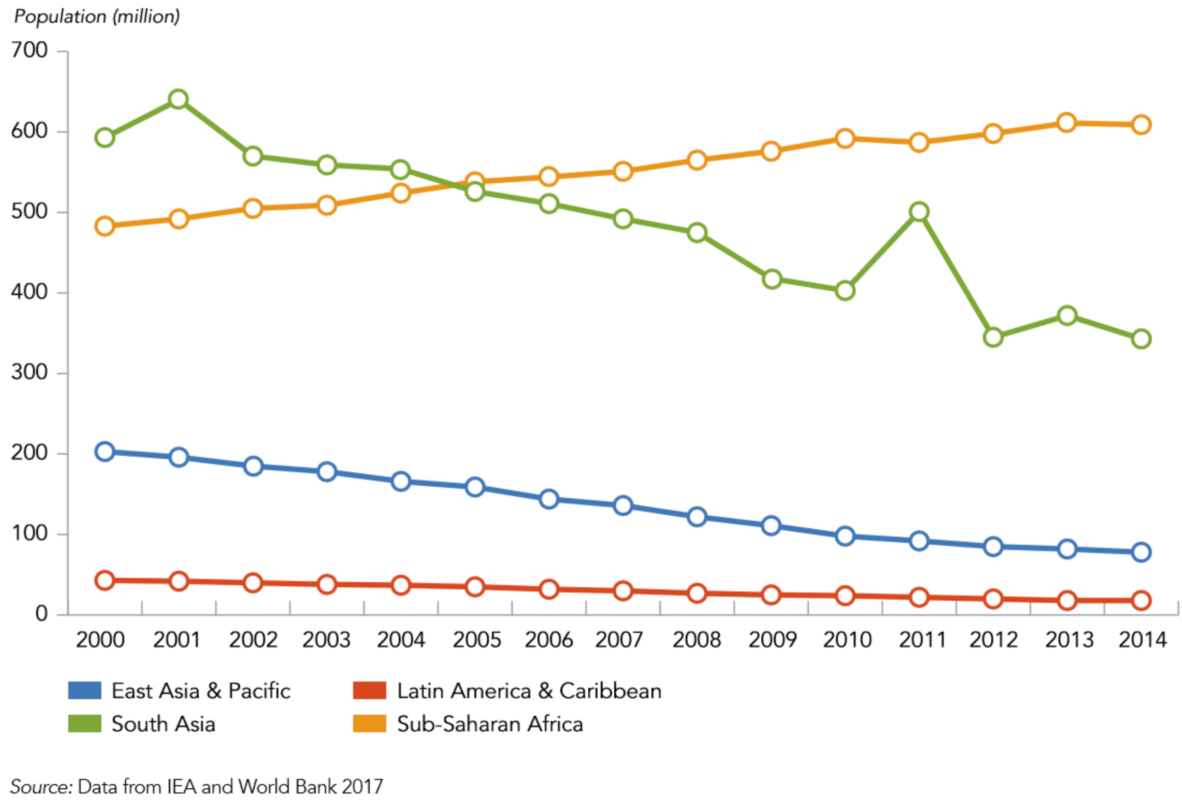



**Figure 1.4: Trends in populations lacking access to electricity**

Developing regions feature the largest energy deficit [122], and this is growing in sub-Saharan Africa due to the fact that energy grid deployments are not on par with population increase. The more energy that communication technologies demand, the less suitable they become for deployment in these regions. Sub-Saharan Africa also has a largely rural [116] and sparse population pattern , putting it at odds with the design of emerging high-bandwidth mobile communication technologies. It is therefore no surprise that deployments of sophisticated generations of mobile telephony and fiber networks are in many cases restricted to urban, more densely populated areas of sub-Saharan Africa [125].

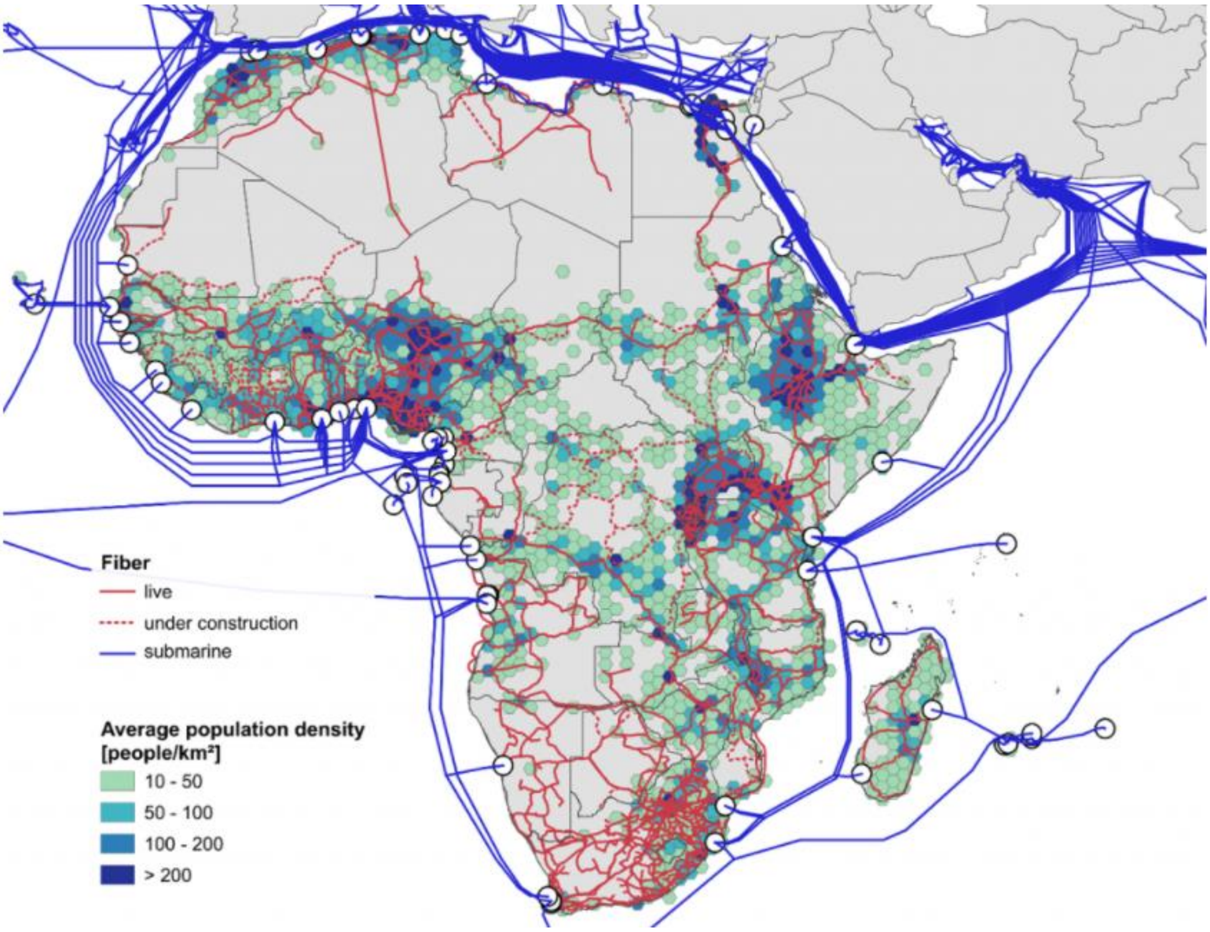



**Figure 1.5: Map showing deployment of fiber links in Africa (source: Network Startup Resource center, Telegeography and the European Commission, 2019)**

Like FM radio receivers, mobile telephones, which are portable and rely on wireless transmission, can thrive in these environments, where users may have to transport them to central points for charging, whereas, in such settings, larger smart appliances such as desktop computers are rendered unusable. Both FM radio and 2G telephony are relatively low-bandwidth technologies, affording increased geographical coverage per antenna installation. This makes them ideal for deployment in areas with sparse populations and relatively poor infrastructure.

#### 1.4.3.2 Challenges to connectivity: Low literacy

Internet-based telecommunication technologies increasingly demand computer literacy, let alone basic literacy, reducing their usability in developing regions. This is compounded by the fact that languages from distanced regions are poorly catered for in modern communication technologies, lacking appropriate alphabets and dictionaries. Therefore, the use of modern communication technologies in distanced regions might not only require literacy, but also the mastery of a foreign language. Sub-Saharan Africa, for example, manifests a literacy rate of 75.51% [107], implying an illiterate population of roughly 300 million people, close to half the population of Europe, before factoring in computer literacy and sufficient knowledge of the languages supported in digital computing platforms.

FM radio and voice telephony have the advantage of demanding the least literacy among electronic communication media, owing to the fact that interaction is mostly oral. It is therefore unsurprising that FM radio and basic mobile telephony are the two most pervasive electronic communication media in developing regions (See Annex A: Core Household indicators). The successes of these two technologies offer some insight into how to connect the global population, which is grappling with the Internet, given that FM radio and voice telephony, at core, expose their end users to basic interfaces despite the fact they are comprised of sophisticated components.

#### 1.4.3.3 Challenges to connectivity: Pricing

According to a 2019 ITU report [129], the price of accessing Internet services lies within a narrow band worldwide, with the price of a basic monthly package of 1.5 GB on a mobile phone costing between $7 and $18 in all markets, and is even cheaper in the least developed countries. The picture becomes less appealing when one considers that the price of the Internet in developing regions is significant in relation to economic indicators such as monthly and GNI per capita income, making the Internet a major expense for many individuals.

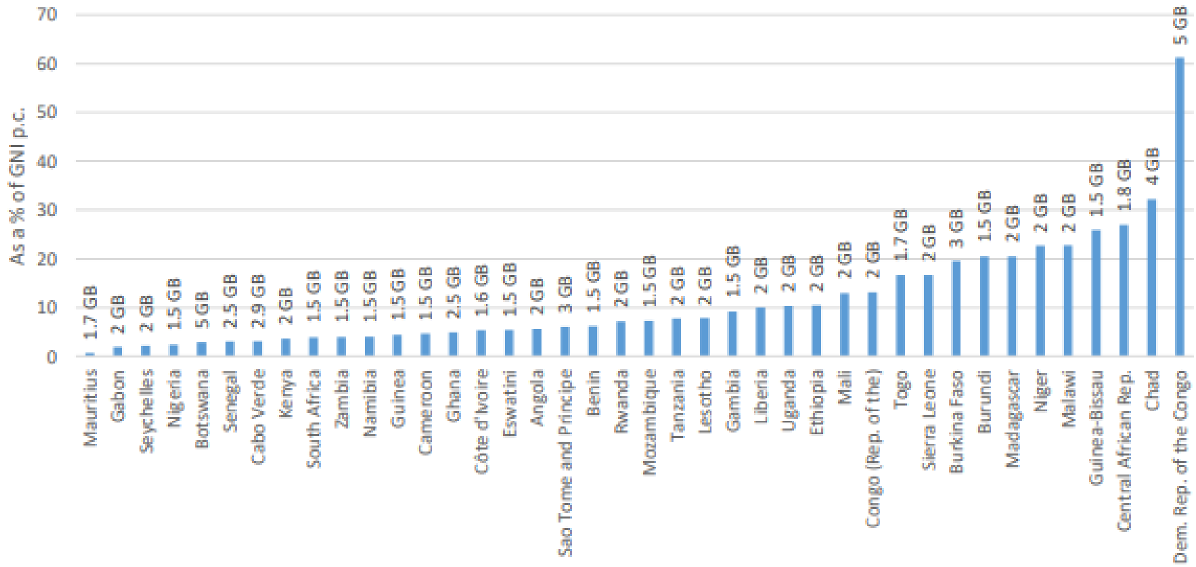



**Figure 1.6: Graphic showing price of a mobile data plan as a percentage of GNI per capita in sub-Saharan countries (source: ITU 2019)**

I later present results from my survey in Pabbo town council in Northern Uganda revealing telecommunication expenditure that did not align with reported earnings, indicating that users were either spending most of their earnings on phone credits, or that there were days when they suspended use of telecommunication services.

It is also observable that telecom companies are making fewer returns on investments in these regions, hence they require the aid of governments in the form of universal service funds, tax waivers and monopolies to offer incentives to extend into these less attractive markets [125]. Any design that seeks to establish communication technologies in distanced regions will have to consider prevailing price constraints, given that they limit the use of technologies that are already widely present in these communities.

I arrive at a design that deploys FM radio technology in a manner that demands much less financing from its proprietors and subsequently from its users. Even though FM radio is free to access, it is expensive to use for communication and this reduces many FM radio “users” to mere listeners. I propose increased participation in FM radio by users who also qualify as listeners, in contrast to a model where FM radio use is dominated by large organizations and wealth owners, whose interests are not in alignment with those of the listeners’ demographics.

#### 1.4.3.4 Challenges to connectivity: Lifestyle mismatch

Many technologies are influenced by the lifestyles of the technologists crafting them. Their introduction in distanced communities is therefore sometimes impeded because these technologies make assumptions about the routines of their users. For instance, I observed that in communities in Northern Uganda where members cannot afford personal computers, efforts were made to provide computers in centralized computing centers, however, this opposes some of the design principles of personal computers. Certain issues, such as personal privacy and preferences, cannot be applied to these shared, communal devices, resulting in a substantially inferior computer use experience.

Another example [40] is the growing integration of fingerprint biometrics in end user devices to increase security. Research shows, for example, that a large number of residents in the Global South engage in labor, mostly casual, that erodes fingerprints, and thereby makes biometric technology less efficient [80,103]. The adoption of technology in distanced regions will always be hampered in a manner commensurate with the designers’ ignorance (or lack of consideration) of the idiosyncrasies of everyday life in these regions. I propose a design that allows communities to contemplate and work with the technologies that are ordinarily provided by an entity far removed for them.

#### 1.4.3.5 Challenges to connectivity: Ownership and control

The lack of control over FM radio and telecommunication networks by local actors greatly hinders the proliferation of electronic communication media in distanced regions. The amount of regulation of these communication technologies is restrictive and keeps distanced communities from owning and operating the communication technologies that they rely on. As Fraser and Estrepo [39] point out, many countries in sub-Saharan Africa, for example, do not have any regulations that clearly define what community FM radio is, and this has resulted in FM spectrum being accessible only to commercial and politically connected entities. During my research, I observed communities in seven different countries who interacted with FM radio regulators and successfully acquired licenses with varying degrees of ease, but in all cases they were assisted by academics and engineering professionals.

Community initiatives around telecommunication and Internet service provision have the potential to foster an organic uptake of digital communication technologies in these communities. Unfortunately, they are heavily fettered by regulations, which in many cases are merely copied from communities that have carefully examined the contributions and detriments of these technologies by observing their use. The possibilities for communities to make their mark on technology, let alone the ability to reconfigure it for their particular needs, are therefore always lost. This is a central consideration in my research, and I seek not only to make the FM radio model manageable, but also reconfigurable by communities in distanced regions.

### 1.4.4 State of FM radio in distanced regions

FM radio is still the primary and, in some cases, sole medium for access to mass information in distanced regions. This is with good reason: it is free to access, listened to on devices costing as little as 1$ that can be powered by cheap zinc-carbon dry cells (available off the grid), and requires no literacy on listener's part. Recent ITU statistics imply that more households have access to a mobile phone than to an FM radio, but my survey in Northern Uganda, and observations during field visits, revealed that many mobile phone owners use their handsets *mostly to listen to FM radio*. For broadcasters, the one-directional nature of FM radio, coupled with low bandwidth requirements, means less investment in transmission equipment when compared to other wireless communication technologies.

The prevailing model of non-institutional FM radio depends heavily on advertising to recoup investment capital and finance operational expenses. Given the areas covered, there is a considerable financial obligation placed on these stations and this has, in effect, two detriments: the skewing of programming towards content with high financial, but not necessarily social returns, and the need for stations to cover large areas to maximize advertisement revenue. Unfortunately, both factors have meant that FM radio is neither ownable nor operable by distanced communities.

FM radio stations tend to be owned by the rich and powerful: politicians, businessmen, religious organizations, and large media conglomerates [79]. This model also aligns with the prevailing use of FM radio as a one-directional medium, funneling messages from the top of the socioeconomic scale to the bottom, with very narrow channels for information to flow in the opposite direction. As I show in section 2.5 in chapter 2, attempts to correct this imbalance by the deployment of community radio have done little to make FM radio stations both ownable and operable by distanced communities. I arrive at a stack that addresses the remnant challenges of community FM radio.

## 1.5 Main contributions

This thesis generates new knowledge that is reusable by practitioners seeking to increase connectivity in distanced contexts. While it focuses on aspects that affect human interaction with technology, this thesis also touches on subliminal aspects removed from the focus of technologists, but which grossly influence the products they design. I list some of the areas where this thesis demonstrates new practice and generates new information below.

### 1.5.1 Licensing consideration for ultra-low power FM transmission

Stations deployed on the RootIO platform held the lowest power licenses in all the countries in which they were deployed. In some of the countries in which RootIO stations were deployed, during the period spanned by my research, there was very little or vague support for low-power FM radio, and small-scale community operators were required to meet the same requirements as large commercial operators in order to obtain broadcasting licenses (see Annex B: Licensing requirements in contexts where RootIO was deployed). By the time of writing this thesis, licenses had been issued for 23 FM de novo stations running on the RootIO platform in 7 countries.

RootIO poses an opportunity for regulators to perceive FM radio at an even smaller scale than is typical in community radio. My experience dealing with licensing authorities helped to revisit existing licensing requirements to see where they inadvertently locked out community low-power FM radio practice, but also to leverage automation to address concerns that have long hampered the liberalization of FM radio.

### 1.5.2 A model for FM space sectorization and delegation of FM frequency management

My thesis breaks new ground in its aspiration towards a model for the sectorization of the FM space. FM regulators are used to handing out FM licenses on a regional and national scale. Many communities hoping to operate FM radio stations Identify their geographic location as an area spanning a radius of a few kilometers; they do not seek regional broadcast ranges. I demonstrate the application of a model that allows an entity to hold FM licenses, and in turn supervise stations that share those licenses and reuse the issued frequency in a manner that does not incur interference. This model allows communities to set up only as much broadcast as they need.

As mentioned earlier, the subtractive nature of licensing meant that stations in Uganda had to complete the same paperwork as commercial stations, “subtracting” the documents that were not applicable to their small deployments. Having observed the contorted processes of applying for broadcast licenses in different countries, it is reasonable to conclude that the prevailing model of licensing fetters the proliferation of community broadcast, as attested to by Bidwell et al [11].

Following my initial study in Uganda and after presenting results to the regulator, one license was granted with a designated frequency in each of three regions, shared out across 10 stations. My work porting the SPLAT [71] RF visualization tool to Java allows communities to easily visualize the reach of their station using basic parameters that are readily available to them or displayed on their transmitter.

This enables communities to plan the setup of their own FM radio stations without having to concern themselves with factors beyond their control. The community can then ask to be included on an existing license, if their setup does not conflict with any existing deployments.

It is a short leap to transform this process into an automated visual monitor, given that all the transmitters I used in deployments had support for RS232 serial interconnection to computers for obtaining and setting operational parameters. Further development in this vein, coupled with the work undertaken in Cape Verde around FM sensing, could result in smarter transmission sites capable of making decisions when they detect interference on their frequencies or automatically correcting to their specified FM frequency in the event of tampering.

### 1.5.3 A realization of the Internet of Radio, facilitated by studio-less transmission and TTS

RootIO presents a model for the Internet of FM radio by facilitating broadcast points for content that are delivered through the Internet. While distanced communities may struggle to use the Internet due to lack of appropriate equipment and literacy requirements, RootIO supports FM technology that is ubiquitous, but leverages modern computing to make the necessity of studios optional, thereby increasing the sustainability of stations with small audiences. RootIO further introduces the use of speech technologies to adapt the mostly written web into audio, making listeners of FM radio essentially users of the Internet.

Making FM radio “Internet-ready” through this approach means that much of the work extending the reach of digital communication networks is not lost on distanced communities, but rather facilitates a use case that they are familiar with. The introduction of TTS allows for the consumption of mostly written Internet information in the oral space where FM radio operates, with minimal human intervention, as is common in Internet-based communication platforms.

### 1.5.4 Increasing interactivity in existing media

The preferred way to fix the flaws of older communication technologies is seemingly to default to the Internet, and this is with good reason. The Internet has equal support for broadcast, unicast and two-way communication, allowing for a broad set of possibilities in the communication space. Features such as adjusting audience sizes, on-demand consumption, and interactivity are very easy to achieve with Internet-based systems that are often resorted to for the modernization of earlier technologies, where conditions permit.

This thesis elaborates pragmatic approaches to fixing the one-way information flow of broadcast media without emphasizing individual Internet access and use. I do not purport to entirely solve this problem, partly because FM radio has a reputation for being monopolized by highly placed entities and individuals, who do not prioritize the use FM radio for intra-community interaction. Where communities had sufficient mastery of the platform, and the social climate was conducive, such as in the Sfantul Gheorghe commune in Romania, examples of novel and hyperlocal use of FM radio were observed, which will be discussed in chapter 5.

### 1.5.5 A hardware and software specification for networked FM radio

In this thesis I propose a software stack that facilitates the syndication of content within, and between, communities. Beyond interconnecting actual transmission sites, the RootIO platform provides tools for collaborative media curation and station administration, within and between, communities to improve the FM radio experience that they own. It is possible for communities to access a wealth of content already catalogued and utilized by other communities, and novel work has been undertaken which allows for communities to set up their stations to “follow” the stations that have active programming, a setup which would see the schedules of the latter replicated in the former

The platform also allows communities to subscribe to podcast sources via RSS feeds and online streams, as well as for content providers to push their content into the RootIO platform and make it available for subscription by interested communities. This collaborative space addresses one of the longest-standing challenges in FM radio: the need for dedicated editorial staff to source and produce high quality content. Not only are these skills in short supply, but where they are available they are not affordable for individual FM radio stations operating at the scale proposed by RootIO.

## 1.6 Chronology of research

My approach begins with an evaluation of a bare-bones implementation crafted in communities, as opposed to a prototype developed in the laboratory and later deployed in said communities. The first prototype of the RootIO project was crafted with communities in Northern Uganda, and at this point it was not clear what the cloud and transmission site technical stacks would look like. After exhausting the trial licenses issued in Uganda, the lessons learned were concretized into a platform that was then trialed in rural Romania.

The deployment in Romania tested both the cloud and transmission site software and hardware stacks and were greatly enriched by communities operating two licensed community radio stations. Given the differences between the communities in Northern Uganda and Romania, the RootIO platform diversified the available technical offerings mostly by increasing support for features afforded by access to Internet.

Finally, the lessons learned by operating RootIO stations over an extended period resulted in designs for viable stacks at the transmission site and in the cloud. In the last phase, I examine the performance of these stacks in a lab-like setting, with an emphasis on the computing platform for the transmission site. Given that the transmission site is often situated in distanced regions, with less-than-ideal conditions for digital technologies, I assess the performance of measures built in to ensure ease of use, recovery, and continuity of service. I also highlight improvements to the cloud platform and their contribution to ease of use and service continuity, as well as improvements which increase the stability of the cloud platform.

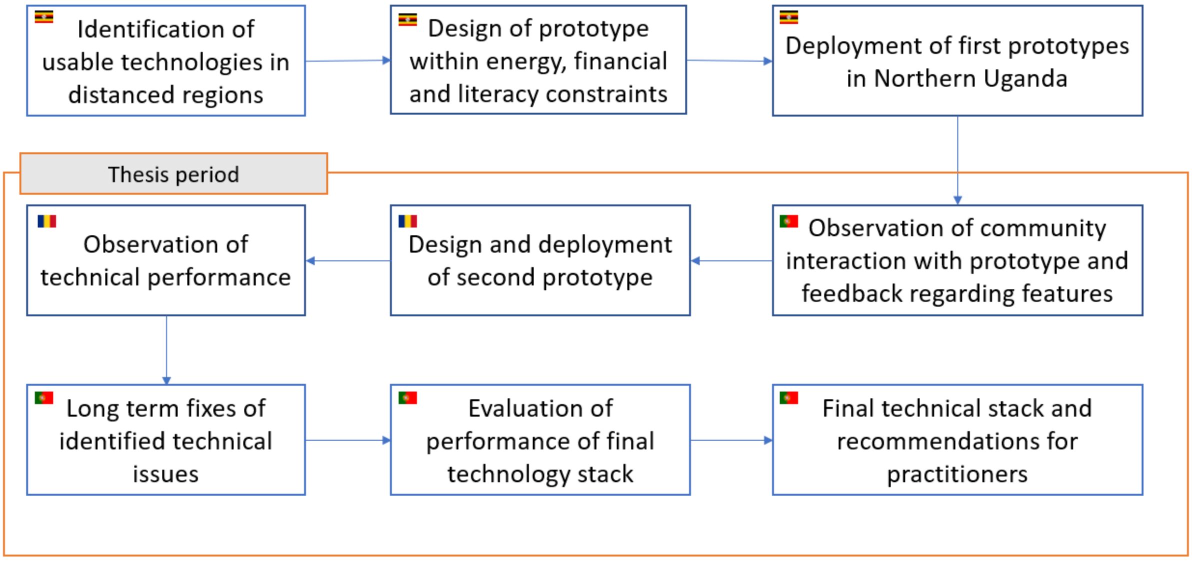



**Figure 1.7: schematic showing the chronology of research with activities during my thesis period**

The change of communities involved is mostly down to the difficulty in obtaining permanent FM radio licenses, which required me to work with communities in which licenses were obtainable within the timeframe of my research. Upon expiry of the initial licenses in Uganda, the application process for permanent licenses was arduous, contorted and barely applicable for the RootIO setup. Subsequently, licenses were granted, but this occurred towards the end of my research. It was, however, possible to obtain long-term licenses in Romania that allowed for proper study of communities interacting with the RootIO platform. Ephemeral licenses were also granted in Portugal, which allowed for an evaluation of technical alternatives, such as antenna designs and hardware constellations.

The variety of contexts, however, enriched the resulting hardware and software options and resulted in a solution that is applicable in varying communities. My research therefore begins by sourcing ideas for an FM radio platform, evaluates the performance of the emergent platform in real world settings and finally culminates in a technical design that is perfected in a lab and considers the needs and constraints in distanced regions.

# Chapter 2

## Review of similar research

This chapter reviews the initiatives that have been attempted to bridge the digital divide and examines their shortcomings. I arrive at a decision on what communication technologies to use and how to best situate them for the maximum benefit of distanced communities.

## 2.1 Introduction

The digital divide [87] provides motivation for many initiatives that seek to increase connectivity in distanced communities. Some efforts have been made on a national scale, spearheaded by governments, to narrow or eliminate this divide. These efforts have mostly been at a policy level where governments try to tilt the playing field in favor of purveyors of communication technologies. In sub-Saharan Africa these have manifested in the telecommunications sector as a mix of universal service funds [121], tax waivers [1] and market fencing by the granting of monopolies [74]. These tools were also employed in the developed regions with success; in sub-Saharan Africa, however, they bring technology closer without necessarily making it more usable.

The work of researchers, in their use of ICT4D, on the other hand, focuses on both crafting new technologies and reconfiguring existing technologies to operate in the space left by policy, economic, social, and physical constraints. My work chiefly belongs in this domain and generates new knowledge in this area, building upon the lessons learnt from prior work. I detail this previously undertaken work below, revealing the lessons learnt from each, and highlighting the space left for further ideation.

## 2.2 Computer Initiatives

Given the success the personal computer has had in developed regions, and the wide range of functions it supports, it is logical to consider it as a tool worth deploying to bridge the digital divide. Many attempts have been made to make computers available, en masse, and within the constraints of distanced communities. While such ideas are noble, the technology is novel and in many cases beyond the comprehension of said communities, thereby lacking community-based champions. I detail the most prominent initiatives that have received both critical acclaim and their fair share of criticism below.

### 2.2.1 The OLPC project

The One Laptop Per Child (OLPC) project sought to make laptop computers so cheap (100$) that every child in the developing region could own one. The stripped-down versions of laptops were procured mostly by governments in developing regions [78] to be issued to school-going children who would use them in their classes and at home to supplement their learning environment. Theoretically, OLPC should have resulted in increased computer literacy by increasing access to computers that children could work with in their daily lives.

An outstanding contribution of OLPC was the orientation towards cheaper personal computers that resulted in offerings such as Intel’s classmate, HP’s classmate and Dell’s Latitude 13. My work draws inspiration from this example, which shows how an intervention in distanced contexts can contribute back to the design of technologies that are consumed in mainstream markets in the developed world. Another example of this pattern is the inclusion of FM radio reception and flashlight capabilities on modern high-end smartphones; uses whose relevance was first noticed in developing regions.

OLPC has two limitations, both of which I consider in my design. Firstly, OLPC deployments were only handled by governments, partly due to the cost implications, and local communities could not self-organize to solicit deployment of OLPC computers within their communities. This is supported by the criticism that OLPC deployments significantly skewed budgets in developing regions and potentially starved the systems necessary to use them, thereby perpetuating the problems they were seeking to solve [58]. This subsequently hindered the organic adoption of OLPC projects in the targeted communities.

Secondly, OLPC introduced a new technology with a preconceived use model, but the product had little compatibility with prevailing educational routines and lifestyles and was susceptible to a lack of communication and energy infrastructure [5]. The OLPC devices therefore gained a reputation as some sort of toy or strange artefact that had to be preserved, rather than used. These shortcomings left little room for users to explore the technology to a point where they could feed back to the design process.

My design proposes a model that allows a single community to launch an electronic communication platform within the finances available in that community. Beyond this, I focus on technologies that communities already own and use comfortably. After witnessing the bureaucracy and paperwork typical of acquiring FM spectrum licenses, I conceived a platform that can support FM spectrum sharing, as well as addressing the concerns that come with lower power FM. This is in a bid to ensure that the deployment of FM radio in distanced communities is manageable by local champions rather than being dependent on experts from outside these communities.

### 2.2.2 Community Information Centers

The concept of community information centers varies from one context to another, but there is a common goal: to create a space where people can access ICT devices that they cannot access within their household. The commonly used name for these centers in developing contexts is “Telecentre”, as if to suggest an estranged or ambassadorial presence of ICT within distanced communities. Telecentres are typically established with funding from governments or international NGOs, entities not based within the benefitting communities, and therefore have a charitable air to them.

In Northern Uganda, these centers are typically powered by solar energy, managed by a local committee and hosted by a local CBO. All fifteen ICT centers that I visited in Northern Uganda received technical support from BOSCO Uganda, a rural church-based IT agency that also doubled as their ISP. This solves two of the challenges identified with the telecentres set up by Grameenphone in Bangladesh: power outages and slow Internet connections [52]. Four of the community centers that I visited in Northern Uganda were also interconnected by VoIP telephony, allowing for calls between the centers.

In Mozambique, ICT centers are collocated with community FM radio stations [86], offering a more holistic attempt at information access. Unlike in Northern Uganda, the community centers I visited in Mozambique relied on grid power and backup was provided by diesel generators, especially because of the collocated FM radio stations. The Mozambique telecentres also featured differing models of connectivity to the Internet: some connect using USB dongles while others have point-to-point wireless connections to ISP transmission sites.

A major achievement of telecentres is the provision of a space that allows for access to personal computers, which are beyond the ownership of many residents, given the energy, financial and literacy requirements. They allow for the use of a sophisticated technology at the bottom of the socioeconomic scale, where market forces prohibit the mere existence of personal computers - home computer access in Mozambique was at an estimated 7% as of 2018 [53]. This intervention, however, does little to increase access to computing – it requires the creation of an artificial environment that is hard for local individual users to recreate on their own.

In Northern Uganda, the telecentres that received Internet from BOSCO Uganda had to assent to rules for accessing Internet service, such as the censorship of adult content, as well as quality-of-service control. The communities welcomed these regulations because they aligned with the general religious beliefs of the community. This lends credence to my thesis that reducing the scale of a communication technology deployment results in costs that are not possible with large deployments, but are at odds with the principles of unfettered access that Servon and Nelson allude to [99], as well as the design principles of personal computing, and the subsequent costs of using computers in ways not anticipated by these principles.

Another negative of community media centers in sub-Saharan Africa is that they are incompatible with the lifestyles of many members of the communities where they are located. On my visits to five centers in Northern Uganda, the users were predominantly young males. This is most likely explained by research that shows that female children in sub-Saharan Africa perform more chores than their male peers [6]. Moreover, these centers are open during the day, when most people would be doing their chores or going to school. This imported model of computer use does not match the lifestyle of those in rural communities where these telecentres are established. The RootIO design elaborated later seeks to use the technologies that people are already familiar with, have access to, and have woven into their daily lives. In so doing, RootIO avoids imposing a lifestyle change on users but leverage their familiarity with prevailing technologies.

Another commonly observed trait in both Uganda and Mozambique is that there seemed to always be several malfunctioning devices that needed repair, but those with the skill for this task could hardly be sourced from within the community. The subject of ICT repair in rural areas is the focus of many researchers [8,56,59], all of whom conclude that the lack of such repair services leads to ICT waste and hampers adoption, especially in communities that are not given over to consumerism, which proselytizes replacement over repair.

This too can be attributed to the fact that a foreign technology was introduced for which these communities were not prepared. The lack of an organic adoption of technology will necessitate continued external support for as long as the communities cannot decipher the technology and innovate around it. The cost of repair in some centers in Northern Uganda was close to the cost of new equipment, given that it was done by a central organization distant from the community. It would, perhaps, have been beneficial to allow more exploratory access to the computers and other equipment, with the potential of developing a group of practitioners capable of carrying out repairs locally.

## 2.3 SMS and USSD apps

SMS is a familiar protocol to many, given its support for man-to-man messaging. It has also been leveraged for man-machine interactions, but its free text and asynchronous nature make it less suited for this latter purpose. USSD is a lesser known but equally present communication protocol designed solely for man-machine communication. The use of USSD was initially only for interaction with core components of telecommunication networks to query and update subscriber status (for example by dialing *100# to interact with a menu as seen in Figure 2.1), but it has been appropriated for communication with third party systems. USSD, unlike SMS, allows for the collection of data in a more structured way and is synchronous, establishing a long-lived session for the entire duration of a transaction, acting as a very low bandwidth application browser. On feature phones with limited Internet capabilities, these two technologies complement voice calls as the available channels for information access.

SMS and USSD technologies have a set of advantages that make them the tools of choice for use in developing ICT4Ds: they are available on all digital mobile phones, are widely used for common basic services offered by telecommunication operators, use very little bandwidth, and do not require the user to have an Internet connection. The two technologies complement each other well: USSD allows for the capturing of structured data, much like web forms, while SMS allows for free text entry. One example of these two technologies being used together is in the civil registration projects that I built to facilitate birth and death registration in Uganda (https://mobilevrs.co.ug) and Mozambique (https://civil.registos.gov.mz/crvs). Data entrants based in communities use a structured form served via USSD to report a new birth, upon completion of which they receive a text message with a summary of the registration. To date, this system has facilitated the registration of 5,000,000 births in Uganda and close to 700,000 in Mozambique since deployment in 2012 and 2016 in the respective countries.

These two technologies have driven some major interventions in developing regions. Mobile banking [50], for example, was until recently built entirely on only SMS and USSD. It is worth pointing out that this service developed out of a grassroots hack in which subscribers would exchange top-up activation tokens by SMS as a form of transferring value [114]; the recipient could resell the token for cash or consume it on their own phone. Mobile banking by telecom companies is now a multi-billion dollar sector in sub-Saharan Africa. The model was adopted by banks that are now also rolling out mobile banking solutions built on SMS and USSD, in a bid to become more accessible to grassroots communities.

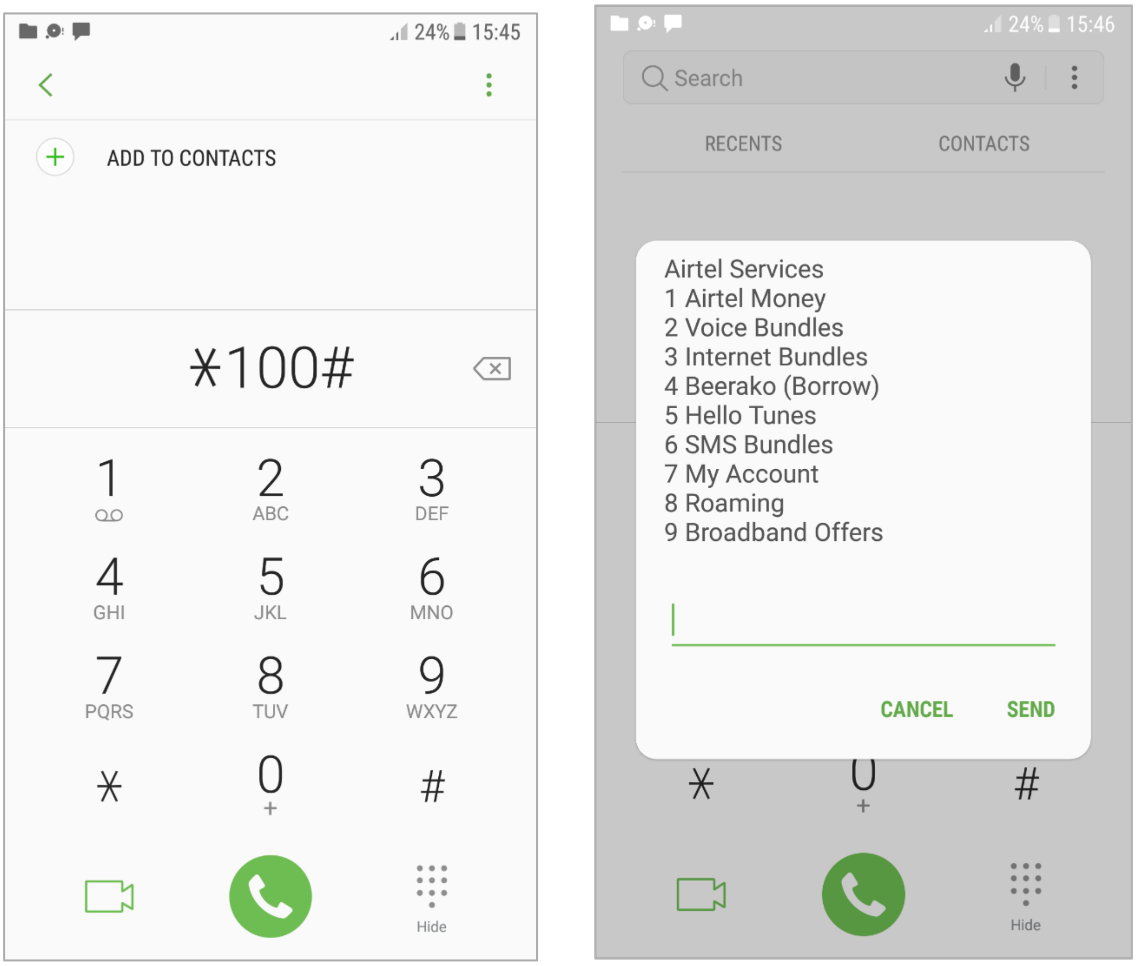



**Figure 2.1: An example of a USSD service menu on the Airtel Uganda network**

Tech unicorns have also attempted to use these two technologies to cross the digital divide: Google rolled out a search system that was based on SMS [3,91] and in my previous work with Uganda Telecom, I worked with a team from Facebook to enable access to the popular social media site via SMS and USSD. Both these initiatives have, however, long died out, most likely due to the very limited bandwidth that SMS and USSD offered, as well as lack of support for rich multimedia. Tech giants are now focusing their efforts instead on increasing Internet access in remote areas [61,120,130].

Researchers have also investigated the possibility of using these two technologies in initiatives spanning the fields of health, agriculture, education and social protection. These technologies are, however, controlled by large telecommunication corporations that are not eager to attend to the needs of small entities (let alone the rural poor) and are therefore only accessible to sizeable entities, such as tech unicorns and international NGOs such as UNICEF, who championed the aforementioned civil registration projects. All solutions that are built on these two technologies require tight coupling with telecom companies, a challenge both on logistical and commercial fronts; indeed the companies have significanly undercut even small, public-good oriented competitors that have relied on their platform [26].

Any attempts to use SMS and USSD beyond basic use scenarios are cumbersome given the limited data transfer rate. The lack of client-side logic handling, similar to what is achieved by the use of JavaScript, complicates crucial aspects such as data validation, making this a less desirable channel. The solution in these scenarios has always been to thoroughly train a few members of the community who then interact with the technology setup on behalf of the larger community. This was the case with various widely-deployed civil registration projects like my own mobileVRS, in which a trained community member would visit a family and use their phone to register a new birth.

My observation of the use of these technologies, even in the most routine scenarios such as mobile money transactions, involve users surrendering their phones to a trained agent who carries out some basic operations on behalf of the user. As such, it should not be assumed that the presence of these technologies in the hands of grassroots communities means that they can be relied on to create a primary communication medium for mass access to media and information. As various researchers have shown [76,90,97,101], it is prudent to plan for intermediaries in the use of ICTs designed for distanced communities.

Attempts have been made to develop apps that offer simplified user interfaces, however, in the background, they use these two technologies for data transfer. Hover (https://usehover.com), for example, provides an SDK that allows for the creation of Android form apps that rely on USSD in the background for data transport. The downside is that these are available only for smartphones and thus face the immediate issues of scarcity of devices and reduced computer literacy. Moreover, from my own work with the RootIO project, I observed that it was only in Android version 8 that APIs for interaction with USSD were introduced in the SDK, but these do not align well with common USSD setups because of the way that sessions are scoped. This shows a disconnect between designers who do not perceive the use of USSD, and users who depend on it for most daily transactions on mobile phones.

In my design, I take note of the success enjoyed by SMS and USSD applications, but also keep in mind the fact that they are not readily accessible to communities for design purposes and are not suitable for the mass exchange of information. I aim for communities to have control over the technologies that they are using. In lieu of USSD and SMS, I employ a combination of IVR and speech technologies as a better adapter to Internet and digital systems. I was able to build interactive IVR applications even in scenarios where telecommunication companies were not eager to provide easier integration with their switching systems, which would be necessary for USSD and SMS applications.

## 2.4 Voice Telephony and IVR

IVR involves the use of audio prompts informing users to perform an action or supply some information. Normally the only supported inputs are numbers, special digits (*, #), and in some cases speech. This area has been explored by researchers in a bid to bridge the digital divide. The appeal of IVR is that it offers an oral means to facilitate interaction with digital technologies. Extending its predominant use in customer care and self-service settings, IVR has been used by researchers to create tools that allow for interactivity in developing regions.

Similar to their attempts in the USSD and SMS area, tech giants have investigated using this channel to bridge the digital divide. IBM's Spoken Web [66] project features a setup in which users call into a parent menu and can navigate sub-sections of a website by listening to audio renditions of the content of the website. As with attempts in the USSD and SMS space, this initiative had little traction: the web is largely visual, which is the motivation behind larger screens in mobile devices. Any attempts at a spoken web result in a substantially diminished user experience, but also an experience that is beyond the scope for which designers are constantly innovating.

Voikiosk [2], Sangeet Swara [110] and Avaaj Otalo [84], on the other hand, use IVR for community dialog in a model where callers leave messages that are then retrieved and responded to by other members of the community, all through IVR. This would be an audio equivalent to web-based Q&A platforms such as Quora®. PhonePeti [65] and RadioMarché [12], which propose IVR as a tool for interaction with FM radio in a model where users call a phone number and, after following prompts, leave a message to be aired on the radio. My previous work with Deutsche Welle Akademie involved building a similar system that allowed for a small, stand-alone computer capable of receiving calls via GSM and storing them under topicalized message categories.

IVR applications, however, leave several challenges unaddressed, which I sought to consider. IVR solutions require integration with telecommunication companies, and, in my experience, this has proved harder in developing regions. As I note in my previous research [25], Telecom companies are observably reluctant to deal with small entities, nor do they accommodate for their various idiosyncrasies. Another negative aspect of standalone IVR systems is their audio nature, which results in slow and sequential access, an undesirable feature, especially when calls are not toll-free.

In my research, IVR and telephony hold central roles. I provide the means for users to input into the repository for recorded content on the station, as well as participate in talk shows, all through interactive telephony. Unlike alternative approaches, which allow community members to leave a message that a presenter then plays on the radio station, in the RootIO setup members of the community input messages directly into the audio content of the station, with little need for any external human intervention. The possibility of abuse is concern that arises here, but the reduced broadcast domain mitigates this problem:

it is easy to identify almost all participants on the radio and resort to local means to resolve incidents of abuse.

## 2.5 FM Radio initiatives

FM radio has unique advantages that make it the ideal electronic communication medium in many distanced communities: it is free to receive on cheap one-directional receivers that consume little energy, and it requires no literacy. Since it uses audio and lacks interaction, it allows for passive listening while carrying out other tasks. Given these facts, it is unsurprising that FM radio is the predominant electronic communication medium in many distanced contexts. The use of FM radio as a communication technology for development is varied. Below, I outline some of the configurations in which FM radio is deployed as a communication technology for development.

### 2.5.1 Development Radio

That FM radio can play a critical role in the development of communities is undisputed. In sub-Saharan Africa, for example, it is common for development agencies to produce audio content addressing thematic areas, for which airtime on existing FM radio stations is procured. This is what I term as “development Radio”. This practice, however, is challenged by the fact that radio stations typically serve a large listenership that may not be interested in messages produced for fringe segments of the audience. At the inception of the RootIO initiative, I met with the UNICEF country representative in Uganda who lamented:

> “We have created very high-quality material, but it won’t get aired”

This was echoed by Farm Voice Radio:

> “We have content that we can demonstrate farmers love, but commercial stations won’t air more than a few minutes of it.”

These programs therefore receive little airtime on mainstream commercial radio stations, to minimize listener churn. Due to the listenership pressure and increased advertisement revenue from multinational businesses, this form of development radio is not especially popular and has been a major factor in the move by international NGOs towards the creation of community radio stations that allow for a more appropriate audience to receive development-themed messages.

The solution to this particular problem is often sought in the creation of community radio stations that should, ideally, be closer aligned with the needs of distanced communities, for whom common topics in development radio are more relevant. As detailed below, the implementation of community radio has left some gaps which I fill with my research.

### 2.5.2 Community FM radio initiatives

Much as with development radio, the community radio scene also greatly varies. Primarily, a community can be defined by a geographic space, but my interaction with communities in Romania and Ireland transcended geographical limits by adding Internet streaming to include members of the community that had immigrated. Therefore, the definition of community became more rooted in shared cultural identity than in geographical location. Beyond community definition, the FM radio stations that I have interacted with in my research vary greatly in terms of geographical coverage, licensing, funding models, programming style and location in relation to administrative and business centers.

According to Fraser and Estrada-Restrepo [40] community radio should primarily:

> - … Reflect and promote local identity, character and culture by focusing principally on local content. Culture is how the people of a community talk about their past and their future. It is what they care about. Like life itself, culture is infinitely variable and constantly evolving. Community culture is also artistic expression through local music, dance, poetry, theatre and storytelling. Local performers are encouraged to go on air uninhibited by considerations of the "professional standards" they may have acquired from mainstream media. Culture is also language, so programming includes the languages of any minority groups in the community.
>
> - … Create a diversity of voices and opinions on the air through its openness to participation from all sectors. Some discord is present in all communities, but the acknowledgement of conflict is necessary for democracy and for democratic communities. Community radio tries to air objectively all sides of a discussion without itself taking sides.
>
> - … Encourage open dialogue and democratic process by providing an independent platform for interactive discussion about matters and decisions of importance to the community. In essence, the core of democratic process is the ability of people to hear and make themselves heard. Community radio provides the forum for that to happen. This is consonant with the decentralization process in many countries that aims to bring democratic decision-making closer to the people concerned. And what is happening at the grassroots level – as portrayed by the community radio programming – can be heard by local government and private institutions, as well as being relayed to policy makers, thus making it possible to design development initiatives that best meet the aspirations and needs of the people.
>
> - … Promote social change and development. In marginalized communities people all have their individual perceptions about their situation, but what is required for change and development is a collective perception of the local reality and of the options for improving it. This collective perception can only be achieved through internal discussions to analyze specific problems, identify possible solutions, and mobilize the appropriate people or groups for action. Community radio provides the perfect platform for this internal discussion.
>
> - .. Promote good governance and civil society by playing a community watchdog role that makes local authorities and politicians more conscious of their public responsibilities. The marginalized and the oppressed normally have no way to complain when authorities take advantage of them, but community radio gives them a voice to air their grievances and obtain their due rights

In my research, I have not encountered any radio stations that meet most, or even some, of these requirements, mostly due to political and financial constraints. As illustrated by Costa [23], many community radio stations are a balancing act between community interest and funder interests, government recognition and funder loyalty, and peddled narratives of the state versus realities of the communities in which they broadcast.

There has been little innovation in the financial models of FM radio, which has stifled the ownership and control of FM radio at a local scale. This, I argue, is due to the aforementioned subtractive approach, in which community radio is arrived at by stripping components out of commercial radio. FM radio equipment is expensive to purchase, and while interventions such as low power transmitters and donations of equipment help to overcome this, there is little that has been done towards a business model for day-to-day operations. Typically, FM radio stations sell airtime and advertisement slots to large corporations, political dignitaries, and NGOs. An FM radio with a reduced audience would not be attractive to many of these entities, and would therefore either have to depend on donations [73] or expand their reach to cover a sizable audience. The RootIO design presents a model for the operation of an FM radio station at a little more than the cost of an individual monthly Internet subscription, thereby relieving the financial pressure around making decisions that define broadcast priorities.

When it comes to technology, the skillset necessary to maintain a typical FM radio station is in many cases missing in distanced contexts.  The three community radio stations that I visited in Mozambique, and one in Cape Verde, were beset by technical difficulties, the solutions for which were beyond the skillset of volunteers or engineers that could be sourced locally. The telecentres hosting community FM radio stations in Mozambique were strewn with non-functional transmission equipment in need of repair, but technicians could only be sourced from the capital, more than 100 kilometers away (and these were, indeed, some of the most accessible telecentres in the country). requires much less technology in these communities, especially when compared to typical FM radio stations: the transmission site bears a very basic FM transmitter that requires minimal configuration. The rest of the functionalities of these stations are hosted in the cloud, providing FM radio as a service through the technologies that people already own and use, thereby shrinking the technical skillset required to operate a low power FM radio station. The situation of most operations in the cloud also allows for experts and other communities to collaborate in meeting the needs of different stations, without actually traveling long distances.

The other area that poses a challenge, but for which there is little support in the research literature, is the licensing of the FM radio spectrum. Community operators are often faced with a tedious process when attempting to obtain licenses for the operation of a low power community radio station. In my experience, this was due to the regulators being far removed from the communities of interest and licensing radio stations at a much larger scale. The community radio licensing scene is varied, with no clear rules on what qualifies an FM radio station as a "community" radio station.

True grassroots broadcast is hard to come by and when practitioners have achieved it, such as in the Tanzanian setting studied by Dunbar-Hester [31], regulators are faced with peculiar setups that they have to closely scrutinize and for which no precedent exists. In my research I take a proactive approach to front the concerns of regulators as well as their respective mitigations, in contrast to negotiating them away. I recognize that truly functional stations are not those that are capable of emitting signals but are among other things viable in the face of regulation.

Some of the FM radio stations I interacted with in Mozambique had no formal license and were precariously operating on non-committal permission from the communications regulator, while in Uganda there were no strict guidelines on how to obtain, or indeed what is, a community radio license. I take the lessons learnt from obtaining licenses from deployments within Europe and sub-Saharan Africa to build a model and tools that allow one to adapt commercial licenses to community use by the delegation of FM frequency to a platform that facilitates sectorizing the FM space into multiple transmission domains for a given frequency.

### 2.5.3 Increasing interaction with existing FM radio stations

Typically, FM Radio stations facilitate the participation of listeners in their programs primarily via voice telephony and SMS messages. With the proliferation of the Internet and social media platforms, FM radio stations are starting to increase their online presence in the form of audio streams and social media handles. This helps to support and prolong debate over several topics that are sampled on the air, but also posted online for continuing debate, for example, on social media pages. With large urban-based stations, this approach favors the urban elite who both have access to the Internet and know how to use it, however, in distanced contexts this is the preserve of the few who can access and use social media.

In a bid to increase interactivity, while bringing in the masses that are offline, some practitioners have relied on IVR tools such as those mentioned previously in chapter 2.4, in order to foster interactivity on FM radio. Some IVR solutions, such as Phonepeti [65], FreedomFone [22] and AFFRI [105], allow participants to call a radio station and leave a voice message about a topic, which is aired later on the station. Other initiatives, such as Sehat Ki Vaani [62], allow users to participate in a live talk show by joining a voice call, but only people on the conference call can listen in on this program.

My design incorporates both aspects but extends each. I include functionality for users to not only comment on a topic of interest but also to use the station to pass on their own information by leaving announcements and advertisements, as well as procuring goods and services. This is akin to creating a post on a social media platform. I also allow for interactive voice calling in a talk-show setting, but the call is broadcast over FM, not mandating listeners to call into a conference, as is the case with Sehat Ki Vaani.

Outside telephony, researchers have proposed the use of specialized devices in the AIR project [102] that allow for the capture of feedback about programs of an FM radio station by recording an audio message on a custom device; the message is then transmitted back to the radio station via a custom wireless network built for this particular device. This design introduces a new device, custom technology, and is very specialized. I propose a design that increases plurality using technologies that are readily available in distanced contexts.

### 2.5.4 Station automation and accessible equipment

Several ICT4D initiatives recognize that financial and human challenges to operating FM radio stations at a community level can be addressed by software that automates some of the routine tasks associated with managing a station. Within the studio, automation software can reduce the amount of hardware necessary, and the number of hours which a human presenter is required for. These solutions therefore offer the potential to increase the sustainability of community FM radio, but do not, on their own, allow for the introduction of FM radio where it is currently unavailable.

An example of automation software is the Airtime™ Pro [131]by Source Fabric, which allows one to source content, create programs out of it, and schedule when it is played. My platform allows this but incorporates telephony to allow for sourcing community content, the upload of text content that is converted using TTS (this can also be sourced from a pre-determined web source) as well as subscribing to podcasts from media houses via RSS feeds. I drew inspiration from Airtime™ but focus on creating an all-round solution that addresses every dimension that communities need to plan for before engaging in discussions with operators to obtain broadcast licenses. The result is a stack that offers basic versions of the core functions of FM broadcast, with communities free to extend and reimagine these functions.

The closest solution to my work is the GRINS [64] software proposed by Gram Vaani for the automation of FM radio stations. The GRINS package, consisting of two computers and a network router, allows for the scheduling of audio content, interaction via telephony, and playing content from streams. The GRINS setup is quoted at a cost of 2500$, on top of other radio station equipment. It also requires knowledge of computing, including manipulation of Linux operating systems, and adds technological components to the deployment. In addition, to use GRINS one needs to have a studio already installed, as well as a working knowledge of the other aspects surrounding the setting up of a radio station.

My proposed setup includes the functions of GRINS, but requires no studio, and no hardware installation past the initial setup of the station, which is optimized to be fitted within an hour if done by two people.[1] RootIO stations in their entirety consist of four main pieces of equipment (a 13kg 22m fiberglass tower, a low power FM transmitter, and a smartphone), costing between 1500€ and 2500€ (depending on whether solar power is necessary or not), and are designed to depend on existing radio

[1] The RF cable termination was the most tedious part of this installation, and we recommend that cables are bought pre-terminated.

stations and content providers for content that cannot be sourced within the community. While GRINS seeks to support FM radio as it is, RootIO seeks to change the model of FM radio in employing a design that relaxes long-held constraints in FM radio, thereby creating a new strain of FM radio that fits into spaces that cannot be operated by commercial radio and existing modes of community radio.

Regarding hardware, few efforts have been made to reduce the cost of some of the most expensive transmission hardware. The only work that we have come across that goes some way to facilitating the kind of connectivity that we wish to see at the grassroots level has been in the form of lower power FM transmitters that are in some cases network enabled, but these are also designed to thrive in a technically rich environment, with access to computers and networking equipment. Based on the findings from RootIO, I concretize the design of a low power transmission ensemble with increased intelligence, and which is capable of operating with increased autonomy in areas with very low bandwidth.

## 2.6 Community Audio Towers

An initiative that deserves mention in this space is the use of Community Audio Towers (CATs). In these initiatives, communities hoist a number of horn speakers in an omnidirectional fashion atop a wooden tower to play audio content from an input source to the surrounding community. The equipment necessary to operate these setups is basic and costst around 300 USD. The use model is also basic, making them very usable in grassroots communities. As Semujju [98] notes, the structure that besets FM radio setups is absent in CATs, making them specific to the communities in which they are operated. He also points out correctly that CATs cover an intra-community communication aspect that is not covered by FM radio stations, including those categorized as community.

CATs do however have their own limitations – everyone listens to the same speakers, and these can be loud at the source while not being heard at a distance from the speakers. There is no way for individuals to opt out of the broadcast and this means for example that CATs cannot be operated at night when many members of the community want some quiet for their sleep. It was also noted in interviews by Semujju [98] that operating multiple CATs simultaneously is particularly challenging, but this spills over into their operation at times when prime content such as news is playing on FM radios stations.

Another challenge is closely coupled with the strength of CATs – because they are basic in technology and operation, anyone can operate them, which is a double-edged sword. On one hand, communities are empowered to create their own media, but on the other hand the lack of a regulatory requirement and framework in many contexts leaves them at risk of misuse, even unintentional, with little opportunity for audit by communication regulators.

Perhaps my biggest deterrent to the use of CATs was out of interactions that I had with the Uganda Communications Commission (UCC) at the start of my research. The UCC was quick to point out that my proposal of a minimal licensed stack was considered by members of the UCC as an alternative that could fill the space that CATs were filling albeit with some discomforts to the regulators. Of recent, the UCC has intensified its crackdown on CATs [], which adds significance to initiatives such as mine that seek to fill this space while addressing the concerns of regulators maintaining usability by the community; maintaining  the affordability of the technological stack and still availing the community control and input mechanisms.

## 2.7 Smartphones and the Internet

The use of smartphones in ICT4D is substantially challenged. To begin with, they are not as prevalent: a 2021 GSMA report estimates that only 56% of all mobile connections in sub-Saharan Africa are enabled by smartphones, of which 64% are older models that support only 3G. Smartphone use requires an ecosystem of social, technological, and economic advances in whose absence the mere prevalence of smartphones may not necessarily result in usage. RootIO uses a smartphone but users of RootIO are not required to. In so doing, RootIO proposes an electronic communication medium designed to work within existing social, economic and technological conditions and with technologies already widely in use.

Smartphones in many uses complement and/or supplement other electronic media without replacing them: owners of smartphones still rely on computers for their document editing and televisions for watching video content over long periods. Even though media producers are adapting and producing shorter content targeted at mobile users, smartphones naturally belong in an ecosystem consisting of other electronic devices. Nemer [77], for example, shows that the proliferation of smartphones in Brazilian favelas does little to eliminate reliance on Community Technology Centers for access and consumption of media. Similarly, Gitau et al [43] expose some challenges to mobile-only Internet access, one of which is that some web-based systems require some access via a computer in order to use them.

The Internet, which facilitates many uses of smartphones, is largely inaccessible in developing regions due to factors listed in chapter 1.4.3, grossly limiting their use in ICT4D. Even where the internet is available, it is not as open and pervasive as in developed regions: efforts such as Facebook zero that have sought to increase internet use by making some sites free to access have resulted in a skewed perception of the internet [115], to the point where some users cannot fully envision the possibilities of the internet beyond social media.

As such, initiatives that use smartphones in ICT4D will in many cases involve a mediator – someone who understands technology, and who is capable of operating a smartphone on behalf of the larger community. This is the case in proposals such as the Open Data Kit [47] and CAM [83,92] that involve a trained individual capturing information from within a community. Sambasivan et al [90] also make the point that some individual and/or private uses of smartphones may be achieved collectively in distanced communities.

The proliferation of smartphones is often used as a proxy to represent their utility within developed regions, but recent events have shown that even in developed regions, distance exists that prevents the use of smartphones. Efforts to curb the spread of the COVID-19 pandemic in Portugal by use of smartphones were deemed impractical after it was revealed that most of the affected elderly population lacked smartphones and/or the necessary skills to use them [126].

It is evident that, while smartphones are and will be increasingly important in distanced communities, they are not a stable enough nor broadly distributed enough platform on which to build solutions aimed at growing the electronic information culture in distanced communities. My thesis does not focus on smartphones even though there are many conceivable use scenarios. I assume a paucity of usable individual smartphones and expose channels that are widely understood in distanced communities, such as basic PSTN telephony.

# Chapter 3

**RootIO: Grassroots Radio for community Input and Output**

This chapter introduces my work on RootIO [24], an intervention geared at facilitating ownership and operation of FM radio stations in distanced communities. I detail innovation in FM broadcast automation, avenues for increased communal participation and efficient sectorization of VHF spectrum for community radio.

My contribution to the RootIO project was informed by my previous experience trying to build communication solutions for the Ugandan population. In previous roles, where I was tasked with appropriating modern telecommunication solutions to the needs of the local Ugandan market, it became clear that many of the smartphone and Internet-based initiatives were receiving less traction in significant sections of the population as a direct result of the aforementioned challenges to connectivity in developing regions.

The basic RootIO idea by Csikszentmihalyi proposed the operation of a low power FM radio station, off of a mobile phone, with the possibility of receiving calls through a soft PBX. I began to investigate alternative situations of digital telephony and interfaces for management of stations as well as integration with modern digital computing advances, all to facilitate community ownership of FM radio. By the end of my research, RootIO is generalized as a cloud-based platform that allows for the operation of low-power FM stations off of low-power, telephony-capable computing devices orchestrated from the cloud through minimal Internet connections. While still fronting analog FM radio and voice telephony, RootIO leverages some of the latest computing advances in cloud computing, IP telephony and machine learning.

## 3.1 Behind the name

RootIO stands for “Grassroots Radio for community Input and Output”. The term “grassroots” in RootIO is a qualifier for the lowest-lying communities on the social, economic, and political scale; I interchangeably refer to communities as “grassroots” or “distanced” but use the former to emphasize communities that are at the bottom of the socio-economic scale, while the latter emphasizes communities distanced from the demographic targeted by modern digital technologies. I consider grassroots communities to be subalterns of distanced communities, and, in this thesis, I explicitly state when conditions, assumptions and results apply to the latter and not the former.

Given the characteristics of grassroots communities, the term takes on a quasi-literal meaning given the dependence of these communities on farming and other activities that are directly in contact with soil. The communities in which the researched RootIO installations were deployed were mostly distant, be it geographically, socially, or economically, from the norm in their countries and from the Internet-capable class, who I consider to be the target of most mainstream communication initiatives.

In many cases, the skills needed to operate and manage an FM radio station are not present in grassroots communities and it is therefore necessary for entities from outside to help in the early stages to get grassroots communities on air. This runs the risk of nurturing a practice known as “astroturfing” [67], derived from the use of artificial plastic stadium surfaces as opposed to actual grass. Astroturfing occurs when entities take on roles in which they represent grassroots communities but have conflicting interests. I observed hints of this practice in RootIO deployments in Northern Uganda and detail measures eventually built into RootIO to ensure increased participation in, and ownership by, grassroots communities.

RootIO places an emphasis on “Input” and “Output” because while many community FM radio projects have been attempted, they offer relatively less accommodation for community input and output. This is usually due to the scale of the station as well as the lack of tools that allow for more community voices on the station. With RootIO proselytizing smaller broadcast spaces, coupled with telephony tools, the chances for individual participation on FM radio are increased significantly, which suggest that input will also be increased. Similarly, RootIO seeks to increases the output of information originating from within these communities and creates space for content and entities that would scarcely be accommodated for by typical FM radio.

## 3.2 Progression of the RootIO concept

The RootIO platform has evolved over the years due to its deployment in several contexts over the past years. Although it would have been ideal to study the platform in a single location, regulatory regimes did not permit this given the excessive bureaucracy involved. For instance, the first request for trial licenses in Uganda yielded licenses after more than a year of waiting and lobbying. The second application similarly took a year to yield licenses and required a lot of technical and organizational information and lobbying. The obtention of licenses in Europe was, however, considerably easier, although though some countries had more stringent requirements than others, and the licenses granted span different times. I elaborate on this in detail in chapter 7.1.1.

Given the delays incurred from the original deployments in Uganda, the subsequent phase of RootIO was deployed and evaluated in Romania. The deployment in multiple contexts provided the advantage of enriching the RootIO solution to cater for diverse contexts. As will be demonstrated, the resulting RootIO platform is versatile and is usable in varying contexts. The lessons learnt from each phase were incorporated into the design for the subsequent phase and evaluated in another context in which the RootIO idea is relevant. My thesis therefore follows these phases and reflects on the state of each of these phases, the lessons learnt, and the design considerations for the next phase.

The first phase was situated in Northern Uganda where four stations were deployed. The goal of this phase was to establish the minimum hardware ensemble necessary to deliver an FM station with broadcast sound quality indistinguishable from that of typical FM radio stations. As such, the emphasis in this phase was on a hardware ensemble accommodating the various components needed for FM broadcast. This phase was also aimed at evaluating the performance of ultra-low power FM transmissions, given that the current 25W proposed by RootIO is significantly lower than the smallest emissions licensed in various locations for community radio.

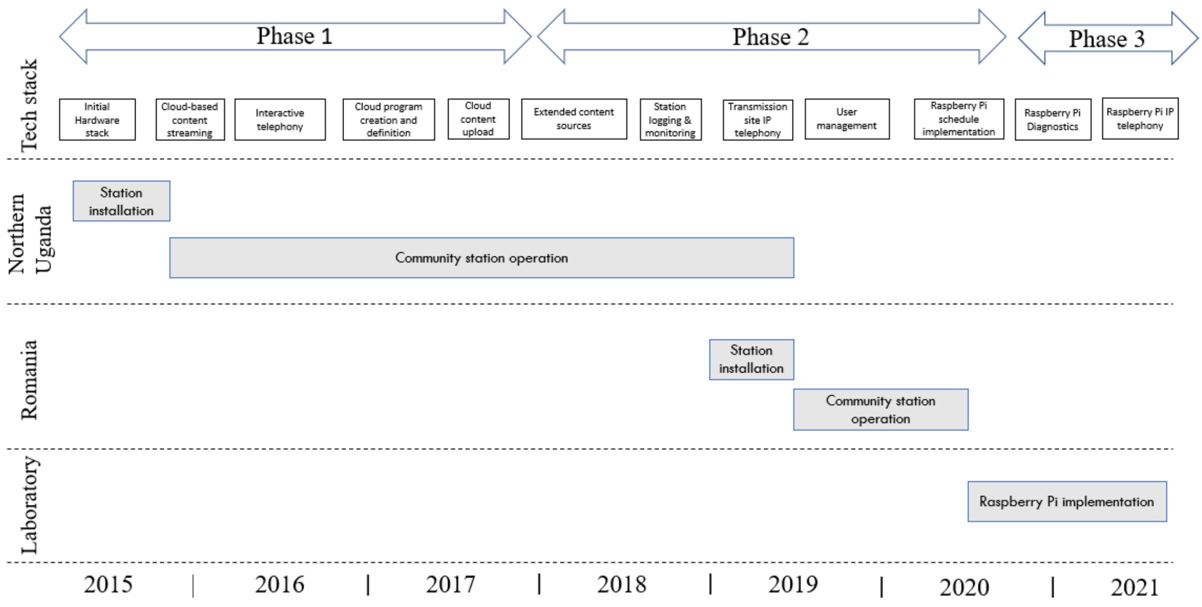



**Figure 3.1: Timeline of the evolution and deployment of the RootIO concept.**

It was realized early in the first phase that there was need for a mechanism to deliver content to the stations from sources without the communities in which these stations were deployed. Another pressing need was for a structured way to incorporate telephone conversations into these stations. The first idea of a cloud-based system to support remote transmission stations was realized with these two features and was used to stream news bulletin files as well as for community members to call into the station and hold conversations.

By the end of the first phase, challenges had been identified in content management and program scheduling. I added functionality to upload media files and associate them with programs that were then scheduled and played on the station. Thus, by the end of the first phase it was possible to play content from the cloud, define programs and communicate over the stations using basic voice telephony.

The second phase saw the installation of two RootIO stations in Romania, four sporadic stations in Portugal, and one sporadic station in Switzerland and Slovenia, respectively. I limit my thesis to the stations deployed in Romania given their long-running status. Among the lessons learned in Uganda was the fact that different kinds of media required different models of sourcing and playback. I therefore added the capability to source content from online streams as well as podcast repositories represented by RSS feeds. The other improvement was in the telephony area that saw the introduction of moderation in telephony interactions as well as IP telephony where Internet at the transmission site permitted. Another significant addition was the introduction of logging of both cloud and transmission site events, resulting in increased visibility of station performance as well as auditability.

In the third phase, I sought to go beyond the limits of the system imposed by dependence on a component of earlier design: the mobile phone as the computing platform for the transmission site. A lot of the failures and delays in rectification in the stations deployed in Romania were down to limitations imposed by the mobile phone though these were features intended for optimization of the envisaged use of mobile phones.

In the third phase I also improved coverage visualization using "SPLAT!", an opensource mapping tool originally written in C++. I ported it to Java, modularized it, shrunk the size of images generated and added transparency to them so they can be overlaid on common mapping platforms such as Google Maps™. The implication is that practitioners can make decisions about the siting of their stations with considerations for characteristics of their transmission equipment. This is crucial given the core proposition of the RootIO setup that stations should proliferate rather than expand, since communities can perceive the coverage of their station over the intended listenership.

Given the insistence on hardware in the first phase and the insistence on software in the second, the third phase involves an optimization of hardware and software in response to considerations pertaining to setup, usability, and regulation.

## 3.3 System overview

The RootIO system implementation can be described as a hub-spoke setup in which a once central server is responsible for multiple transmission points, and each transmission point is controlled by only one central server. A RootIO instance is a setup in which one or more transmission points are connected to one functioning cloud based RootIO server. The distribution of logic between the cloud and local transmission point helps to promote the advantages of cloud computing and modern Internet technologies, while maintaining a local content cache and logic to ensure continuous broadcast in areas where Internet connectivity is not reliable due to cost and infrastructural challenges.

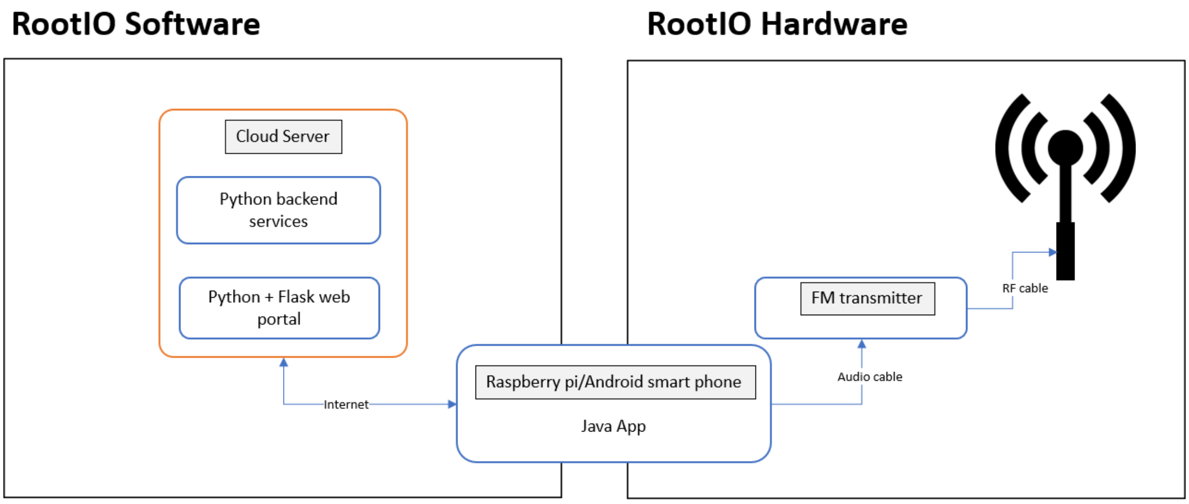



**Figure 3.2: Schematic showing the link between the different RootIO software and hardware components**

As can be seen from the schematic in Figure 3.3 below, the interaction between the cloud platform and the transmission points is via telecommunication networks, whether through data channels or traditional PSTN voice and SMS channels. It is necessary to inform the reader that the use of PSTN in the schematic denotes only voice calls and SMS messages made over 2G/3G/4G networks as opposed to the Internet, that is also facilitated by PSTN networks. By virtue of providing the connectivity between the cloud, the transmission points and users, telephony networks play a pivotal role in the RootIO ecosystem, and their offerings significantly affect what is possible in a particular community.

Below we take a detailed look at the individual components of the RootIO ecosystem and the role they play in facilitating broadcast and the exchange of information at a lower level.

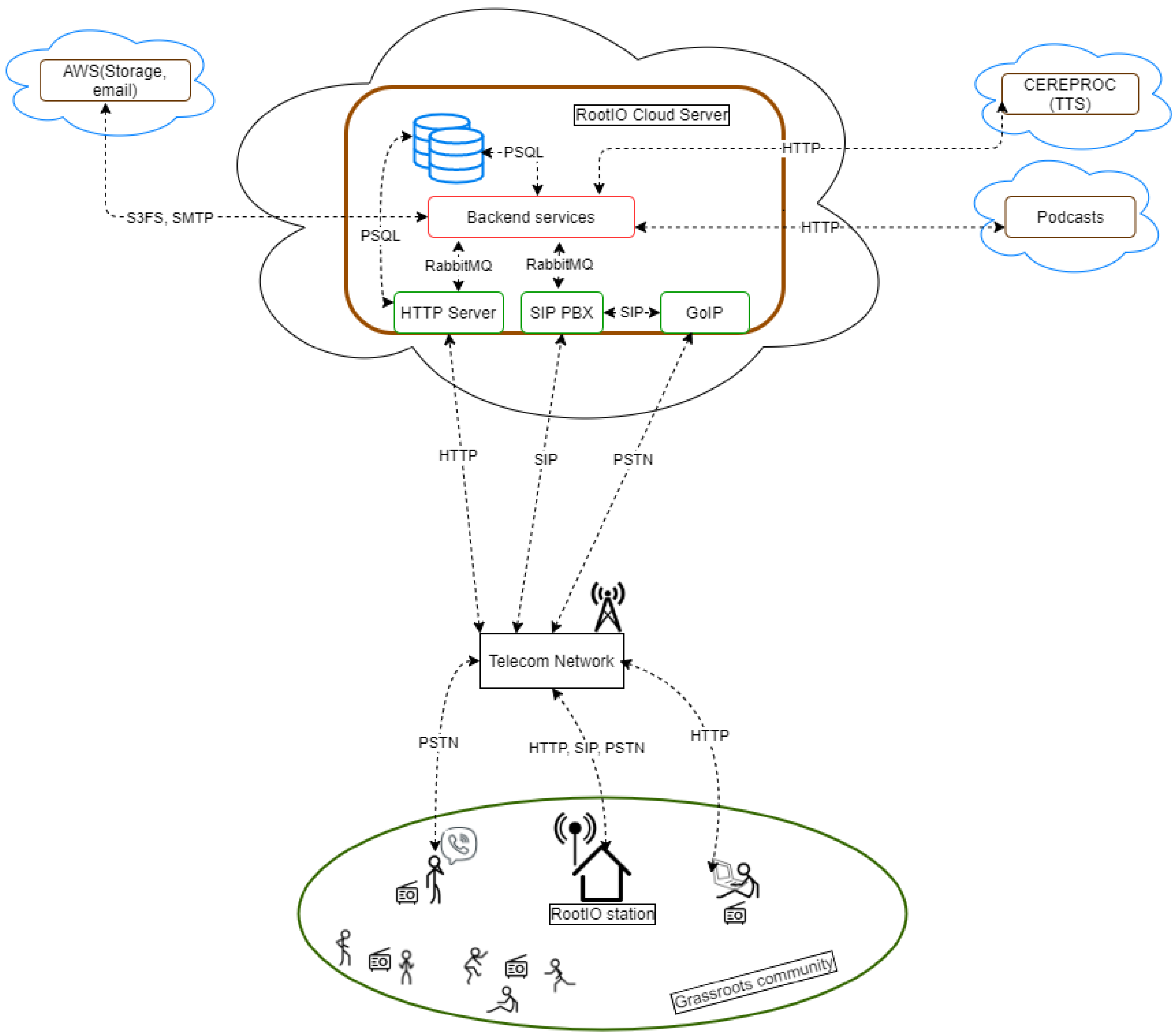



**Figure 3.3: High level RootIO instance architecture**

## 3.3.1 The cloud platform

The cloud platform is the brain for all the operations pertinent to stations associated with a particular RootIO instance. The cloud platform is theoretically the single point of interaction with the different

RootIO transmission points, and therefore hosts functionality for configuring and orchestrating the transmission points. The current setup is hosted in the cloud by Digital Ocean®, on a server that runs Ubuntu® version 16. The functionalities offered by this server include:

A web portal: This is implemented in Python using the Flask web framework (initially by developer Josh Levinger), served by an Apache® web server. It allows for functions that include:

- Creation of logical networks (groups of stations) and their associated stations.
- Definition of and upload/subscription to content from different sources (Streams, Podcasts, Media Files).
- Creation and administration of users, and associating them with their respective networks.
- Creation and scheduling of programs, encapsulating content from different sources.
- Monitoring diagnostics and activities of the various transmission points attached to an instance.
- Configuration of the different parameters that define the cloud platform's nature of communication with a particular transmission point (telephony channels, frequency of synchronization).

The web portal offers a single point for the setup and management of transmission points. Some parameters and settings are applied on the cloud platform, while others are passed on to the transmission site to delegate station operations to a minimal but autonomous computing platform. The code for the web portal is available online at: https://github.com/rootio/rootio_web.

A database server: The database server is a central repository for all information pertinent to the RootIO instance. All the information that is entered and manipulated via the web portal is retained in this database, as is status and activity information that is periodically sent up from transmission sites. The database also serves as an information repository for background services tasked with sourcing content from the various sources specified through the web portal. The current database is built on a PostgreSQL 9.0 database server.

A content repository: The cloud platform also hosts a content repository for all the audio content that is streamed to individual RootIO stations. In reality, the audio files are stored in an AWS S3 bucket mounted as a disk on the cloud server through S3FS. Effectively, all files uploaded to the cloud platform are backed up in a cloud location and would not be lost even if the actual server were to fail irreparably.

A telephony soft switch: The soft switch allows for the cloud platform to make calls to the transmission site; the calls are in turn broadcast to the community served by the transmission site. Where the Internet allows, these calls are via a direct IP connection, otherwise via a PSTN network. Upon acceptance of the call by the transmission site, an event is propagated to the RootIO cloud software to perform a relevant action such as playing a media file or calling a phone number belonging to someone in the community. To this effect the soft switch provides an API and broadcast for events around the origination of a call, acceptance or rejection of a call, start and end of media playback, and the keying of DTMF digits. The current soft switch is implemented using FreeSwitch® version 1.6.

A GOIP SIM bank: GOIP devices add the possibility to adapt 2G telephony into IP telephony, allowing the cloud platform to use physical SIM cards to originate and receive calls and SMS on a mobile network. Though denoted as residing in the cloud, this is in a sense that is more logical than practical In practice, in fact, GOIPs need to be stored in a location where they receive cellular network signal, and datacenters proved to be a less than ideal setting given that they are always heavily enclosed. As such, these GOIPs are, in many cases, located either within the community hosting the station or somewhere else within the same country, while the cloud server may not be. The fact that they communicate with the cloud server over IP networks but never directly with the transmission point makes them logically closer to the cloud, hence their grouping with cloud components.

Background services: The cloud platform hosts three long running RootIO background services, namely: the program scheduler, the telephony menu, and the podcast download service. The scheduler uses information stored in the database to create a schedule of programs and initiate their broadcast on the relevant stations. The scheduler service leverages telephony to call up transmission sites and play pre-recorded content once the station answers. Thereupon, the scheduler service will log the content played for which station, and at what time, as well as logging if the attempt was successful or not.

The download service on the other hand is only responsible for downloading podcasts periodically from the various sources defined by supplying the URL to the RSS feed for a particular podcast program. In chapter 3.4.3, I go into further detail about the process of podcast downloads. Developments have been made by independent researchers to enable subscription to RSS feeds for text content that is then synthesized using TTS and eventually played on RootIO stations [94]. This is, however, outside the scope of my thesis, but it does present an example in which communities have innovated around the RootIO platform.

The third service is a telephony menu that facilitates interaction with the station by members of the community. This service listens to events coming into the telephony system and processes them using a predefined dial plan that makes use of an IVR menu. In chapter 3.7.2, I detail the role that the telephony menu plays in facilitating community messaging and participation in talk shows. These services are developed in Python 2.7 and their implementation is available at https://github.com/rootio/rootio_web/tree/master/telephony.

## 3.3.2 The transmission sites

The transmission site features a low power FM transmitter (25 Watts) and a programmable device capable of playing audio files, communicating over the Internet, and receiving calls and SMS messages.

The obvious choice for this smart device was an Android smartphone, given that it supports all the above-mentioned functionalities out of the box, but as I will later demonstrate, it was justified to invest time into building PSTN telephony, IP telephony and media play capabilities into Raspberry Pi computers that demonstrate better performance and stability over longer periods.

A plastic casing hosts the smartphone and low-power transmitter as well as the audio cable that connects the phone to the transmitter. Also included is a charging mechanism for the phone and a powering mechanism for the transmitter.


**Figure 3.4: Image showing earliest RootIO transmission site devices (Android phone and Transmitter) housed in actual bucket**

The ensemble is designed to be powered by a 12V DC energy supply, which can be easily sourced from solar setups, but is adaptable to grid power where necessary and/or possible. The RF signal generated by the FM transmitter is fed by a 50Ω RF cable into an antenna hoisted 15-20 meters above the ground for emission to the surrounding areas. Below we delve into the specifics of the different components and their function at the transmission site.

Low-power FM transmitter: The low power transmitter serves as an interface for the communities to leverage the conveniences afforded by the RootIO setup. By broadcasting content over FM channels, communities around the transmission site perceive RootIO stations as ordinary FM radio stations, despite being significantly different behind the scenes. The initial transmitter used in prototyping was the CZE 15A from CZH FM, though this was never deployed in communities. Three different transmitter types have been deployed: a CZE T-251 from CZH FM, the Mozart 50 from Digital Broadcast and a CYBERMAXFM+ SE+ 25W from PCS Electronics. All these transmitters have differing capabilities with different price points as compared below.

| Feature | CZE 15A | CZE T-251 | Mozart 50 | CYBERMAXFM+ SE+ 25W |
|---|---|---|---|---|
| **Cost** | 158 USD | 200 USD | 2100 USD | 550 USD |
| **Power range** | Fixed 15W | Adjustable to 25W | Adjustable to 50W | Adjustable to 25W |
| **Audio inputs** | 3.5mm Aux | RCA | RCA, MPX, SCA, AUX, Digital AES/EBU | XLR, MPX, USB, RCA |
| **Audio options** | Mono, Stereo | Mono, Stereo | Mono, Stereo, Audio Equalizer | Mono, Stereo, Audio Equalizer |
| **SWR protection** | No | yes | yes | yes |
| **Computer interface** | No | RS 232 | RS 485, IP networking | USB |
| **Telemetry interface** | No | No | Email, SNMP, Web portal | RS 232 |
| **Network interface** | No | No | WIFI, Ethernet | Ethernet |
| **Form factor** | 6” * 4.8” * 1.7” | 8.2” * 6.8” * 2.3” | 1U, 19” depth | 1U, 5” depth |

**Table 3.1: A comparison of the key factors across the 4 transmitters used in the RootIO project**

As can be seen from Table 3.1 above, as prices increase so do the options available. For the most part, the more expensive transmitters feature increased audio input options, audio enhancements, and connection options to computers and IP networks. I observed, however, that these features were not easily usable because they required advanced technical skills and equipment. The CZE T251 strikes a good balance between ease of operation, useful features such as SWR protection, audio options and computer interconnection. The form factor of the CZE T251 also makes it more suitable for field deployments as opposed to the rack forms of the Cybermax and Mozart series transmitters that belong indoors in a rack.

Android smartphone: The smartphone adds a brain to the transmission site, capable of acting with a degree of autonomy and being capable of dynamically making decisions based on logic, defined in a pre-installed app, as well as configuration information fetched from the cloud server. The phone plays audio that is channeled from the earphone jack into the audio input of the transmitter for subsequent broadcast. In addition, the app installed on the phone searches for calls coming into the phone, comparing the originating numbers with those on an access list, thereby answering or rejecting calls accordingly. When a call is answered, the phone pauses any media that it is playing and switches to piping the audio of the call into the FM transmitter.

Besides these two core functionalities, the phone is continually sending details about its status (network, storage utilization, network connectivity, logs of media play, lists of calls and messages received and current location) as well as requesting information from the server (whitelisted phone numbers that can call in, playlists created out of the media on the phone, a list of programs and their associated content to be played on the phone, configuration information for how to connect to the cloud, soft switch for calls, and information on how frequently to synchronize with the cloud platform).

Given that the phone can play programs constructed from playlists of content on the phone itself, as well as receive phone calls, it is capable of running programs offline and allows for communication from within the communities for extended periods with no connectivity to the cloud platform. The current implementation runs on Android phones running Android 8 (Oreo) or later. The major constraint is that programmatically answering phone calls is only officially supported in these particular Android versions, though it was previously possible via unofficial workarounds until Android 4 (Jellybean). I later demonstrate that switching to Raspberry Pi as the computing platform of choice results in a more predictable platform for future development.

The Antenna kit: The antenna serves to emit the broadcast to the surrounding communities, and to this effect it should be hoisted atop an elevated mast with sufficient clearance of dense obstacles that might significantly degrade the quality of the signal that is going out. I have used different antenna designs, both off the shelf as well as some antennas built from scratch. While any FM frequency antenna will work, considerations have to be made for the mast hoisting the antenna.

**Figure 3.5. Images showing a steel tower with side mounted T-shape dipole antenna in Uganda (left) and fiberglass antenna with top mounted monopole antenna with ground plane in Ireland (right)**

In areas where stout metallic towers were used the preference is for dipole antennas mounted some horizontal distance from the tower, whereas in contexts where telescopic fiberglass poles were used, the preference is for monopole antennas mounted in line with the tower, or dipole antennas without an extending arm mounted along the tower.

The antenna is connected to the transmitter using 50Ω RF coaxial cable, with preference for low-loss cable suited for outdoor use. Existing setups make use of LMR240 and h155 low loss cables [132]. The transmission lines feature coaxial surge protectors to eradicate the risk of lightning strikes damaging the FM transmitter. For my installations I used both dipole and monopole antennae with differing characteristics, but this analysis is outside the scope of my research, save for polarization and gain characteristics that I relied on for coverage modelling, which is detailed in chapter 6.7. I placed emphasis on the combined performance of the transmission cable and antenna as a lumped system by measuring the SWR of the entire setup. In all cases I was able to make adjustments that brought the SWR closer to 1 and below 1.5. This was achieved by better cable crimping, careful antenna length adjustment, securing the cable from movements likely to result in loosening, avoiding kinks in the RF cable, and making sure the antennae were clear of obstacles.

<u>Power equipment:</u> The contexts in which RootIO stations are deployed are the most likely to have decreased access to energy grids. As such, the design that I propose is for stations to be able to run entirely on solar energy. The stations in Uganda ran entirely on solar energy for which equipment was dimensioned using the following formulae:

The daily wattage of the system is given by:

$$P = 24 * IV * L_B \tag{3.1}$$

Where $I$ is the current draw in amperes and $V$ is the system voltage and $L_B$ is a loss factor to cater for battery losses. An $L_B$ Value of 1 represents a perfect battery, while every percentage loss is represented by an additional 0.01.

The daily amperage is given by:

$$I_{HD} = \frac{P}{V} = 24 * I * L_B \tag{3.2}$$

It is often a requirement that the setup can retain charge for a certain number of days due to unfavorable weather, or while pending repairs for damage that prevents charging. The total charge to be stored by batteries is given by:

$$I_B = D * \frac{I_{HD}}{F_I} \tag{3.3}$$

Where $D$ is the number of days for which charge should be held and $F_I$ is depth of discharge on a scale of 0 to 1 with 1 representing full discharge. The number of batteries is given by:

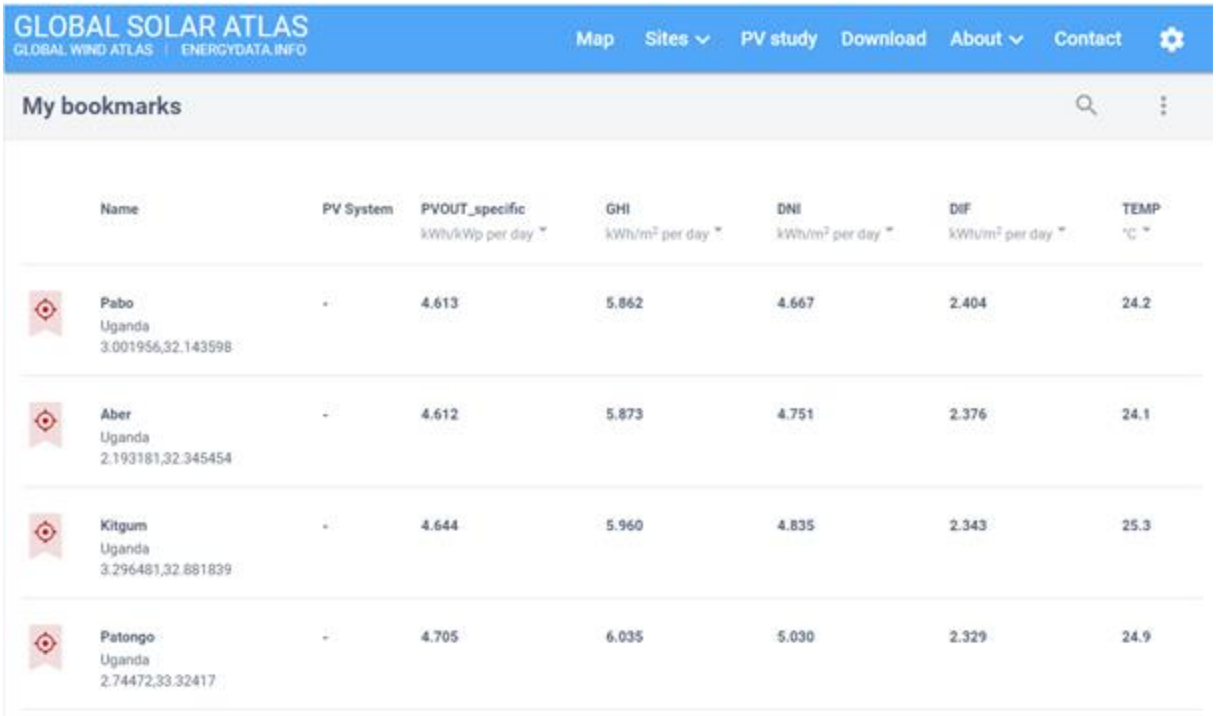
$$N_B = \frac{I_B}{I_{AH}} \tag{3.4}$$

Where $I_{AH}$ is the capacity of the battery in ampere hours. The number of solar panels needed to fulfil the demand of this system is then given by:

$$N_s = \frac{I_{HD} * V_p}{P_p * G} * W \tag{3.5}$$

Where $V_p$ is the nominal solar panel voltage, $P_p$ is the watt rating of the panels used, $G$ is the solar global horizontal irradiation and $W$ is a weather factor to compensate for deviations in weather that negatively impact the amount of solar radiation received.

The batteries used had a listed loss rate ($L_B$) of 2%, and the combined system had a power rating of 61W ($P$). The desired number of days ($D$) to hold charge was 2, given the difficulty in obtaining repairs in northern Uganda. The depth of discharge ($F_I$) was considered as 0.5. with the batteries having a storage capacity ($I_{AH}$) of 210 Ah. The solar panels used had a nominal voltage ($V_P$) of 16Vand a watt rating ($P_p$) of 130W. The global horizontal irradiance ($G$) for locations in northern Uganda ranged between 5.8 kWh/m$^2$ and 6 kWh/m$^2$ per day (Figure 3.6). $W$ is a scale factor that caters for weather variations and accumulation of dust, with 1 representing a perfect setup, although in my case this was set to 1.5. Another potential detrimental factor is loss due to not observing the recommended tilt angles, on which there is plenty of literature, as reviewed by Jain and Lalwani [57]. I observed, that technicians had no regard for tilt angles and orientation, but simply worked with the orientation of the roof on which they installed. Fortunately, close to the equator the loss due to a wrong tilt angle is at its lowest.




| Name | PV System | PVOUT_specific kWh/kWp per day | GHI kWh/m² per day | DNI kWh/m² per day | DIF kWh/m² per day | TEMP °C |
|---|---|---|---|---|---|---|
| Pabo Uganda 3.001956,32.143598 | - | 4.613 | 5.862 | 4.667 | 2.404 | 24.2 |
| Aber Uganda 2.193181,32.345454 | - | 4.612 | 5.873 | 4.751 | 2.376 | 24.1 |
| Kitgum Uganda 3.296481,32.881839 | - | 4.644 | 5.960 | 4.835 | 2.343 | 25.3 |
| Patongo Uganda 2.74472,33.32417 | - | 4.705 | 6.035 | 5.030 | 2.329 | 24.9 |

**Figure 3.6: Picture showing the global horizontal irradiance values at the transmission sites in northern Uganda (source: Global Solar Atlas)**

Enclosure: For protection against environmental elements and tampering, the transmitter and mobile phone are enclosed in a plastic container designed with ventilation and power inlets. In earlier settings, the preference was for containers that are readily available within these communities, such as industrial plastic buckets for common products such as paint and detergent. The form factor of transmitters used in later deployments, however, necessitates rack-form equipment cases.

Given that in most of the deployments the equipment was kept indoors, little emphasis was put on making the enclosure environment-proof. I observed, however, that communities always came up with designs to keep the equipment safe from tampering and environmental elements once the necessary considerations around electronics and RF cable were explained to them.

| Equipment | Function | Alternatives | Notes |
|---|---|---|---|
| Low-power FM transmitter | Modulates audio content onto an RF signal in the FM radio frequency range | CZE 15A, CZH T251, CYBERMAXFM+ SE+ 25W, Mozart 50 | Initially, the CZH transmitter was deployed but reluctance by the manufacturer to supply certification conforming to broadcast standards resulted in the switch to PCS Electronics transmitters. These came with a CE certificate and test results for the relevant standards from the manufacturer, an acceptable compromise for the regulator in |

| | | | |
|---|---|---|---|
| | | | Uganda. The form factor of the CZH transmitters was appropriate for the bucket enclosure, while the PCS transmitters were the major factor in switching to rack-form cases. |
| 50Ω RF cable | Conveys RF signal from the transmitter to the Antenna for broadcast | H155, LMR240, RG8, RG58 | The initial deployments in Uganda used both RG8 and LMR240 cables. The RG8 cable featured higher loss and was heavier and more rigid, demanding more trunking work, while the LMR240 cable was lighter and had lower loss. With the switch to fiberglass poles, the LMR240 cable was still considerably heavy, and I preferred H155 (PE), whose signal attenuation is closer to that of LMR240, albeit at a lower weight per unit length. |
| FM broadcast Antenna | Propagates FM signal over surrounding area for reception by individuals with FM receivers | FM User DP 100, SIRIO GPA 66-108 monopole, SIRIO DP-100, fabricated half-wave dipoles. | The initial deployment used an FM User DP-100 dipole Antenna and an on-site fabricated Half-wave dipole Antenna. Subsequent deployments used a mix of monopole and dipole antennae. The key factor was whether fiberglass or metallic towers were used to hoist the antenna. |
| Antenna tower | Hoists the FM broadcast Antenna over the ground to facilitate propagation of the FM signal with reduced obstacles in the Fresnel [68] zone. | Custom-fabricated 15M steel tower, Spiderbeam 22M fiberglass pole | In initial deployments, custom-designed steel towers were fabricated but, given the labor and cost requirements, the preference was for prefabricated fiberglass towers that are lighter, and require less manpower and skill to install. |
| Android Mobile Phone | Plays media and phone calls into the audio input of the FM transmitter for broadcast. Hosts app | Samsung Pocket (GT S5303), Samsung A6+, Samsung A20e, Samsung A10, | Initially, Samsung pocket phones were preferred because of their price point and their support for call answering. Subsequent models with later versions of Android lacked support for auto-answering |

| | | | |
|---|---|---|---|
| | which runs station according to defined schedule and content sources | Nokia 3.1 (TA-1063) | phone calls, a feature reinstated in later versions, starting with Android Oreo, which allowed the use of more powerful phones. |
| Solar Equipment | Equipment for harnessing, storing and supplying solar energy to the station | six solar panels totaling 810W at maximum sun exposure, two 12V 100Ah dry batteries in parallel, two charge controllers (each connecting three panels to one battery) | In Uganda, where energy grid access is scarce but sunshine is abundant due to location along the equator, solar energy is a worthwhile investment. In European contexts, however, the capital expenditure is not justified because grid energy is more widely available and relatively much cheaper. |
| Enclosing container | Encases all the station equipment for protection against environmental elements and tampering | Industrial Water-tight Polypropylene bucket, Gator 2U rack case | The form factor of the transmitters used in the initial setup allowed the use of buckets. The subsequent use of rack-size transmitters influenced the choice for rack cases. |

**Table 3.2: Equipment hardware options at RootIO transmission sites**

## 3.4 Functions facilitated by the RootIO cloud platform

The overarching goal of RootIO is to situate FM radio stations in distanced communities in a way that allows for these communities to use these stations to send and receive information. To this end, the RootIO cloud facilitates functions that allow for the operation of these stations, with increased accommodation for content created within the community hosting the station. The cloud platform also allows for the performance or automation of other tasks typical of FM radio stations, but trades perfection for convenience where the marginal gains of adhering to standard FM radio practice demand substantial sophistication. In this section, I elaborate the different core functions that allow for the operation of radio stations on the RootIO platform.

While many operations are performed on the RootIO platform, many are only done once, when first setting up a station or signing up. The routine operation of FM stations revolves around sourcing content, creating programs that encapsulate this content, and adding those programs to a schedule. These latter three functions, coupled with the background scheduler that orchestrates programs on the cloud platform, are the most critical, and my thesis is concerned with guaranteeing their performance.

### 3.4.1 Signing up

I begin with user management because the first actions that one performs belong in this area. To access any of the functionalities of the RootIO platform, one has to sign up to the platform by clicking the "sign up" button which is one of the options presented when one accesses the platform before logging in. Upon supplying an email address that serves as the username, and a password, the user receives an email asking them to verify their account by clicking on a specified link, after which they can use their credentials to log in.

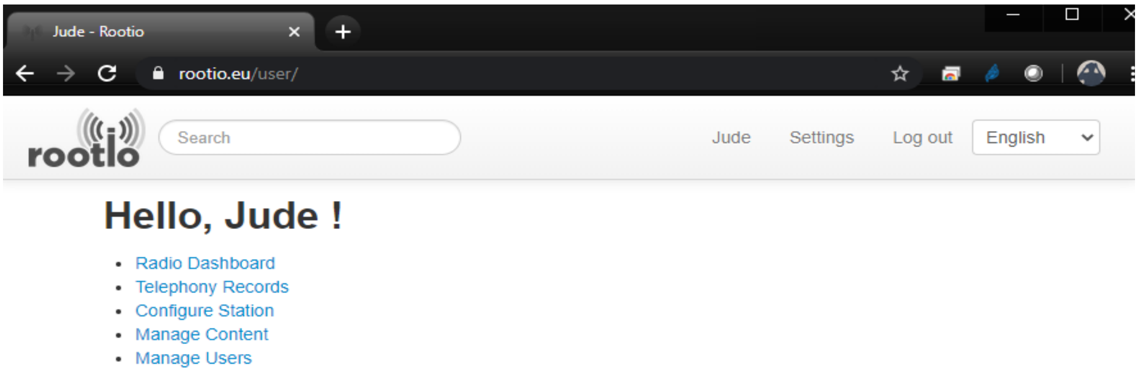



**Figure 3.7: Landing page of the RootIO platform showing the main menu**

### 3.4.2 Creating a network and stations

FM radio stations are grouped into networks. This was necessary because the operation of stations at the scale proposed by RootIO results in the sectorization of a large broadcast area, which typically would be covered by one FM radio station. As such, RootIO stations are most likely to be set up in clusters of more than one transmission point, constituting a network of stations.

To create a network, one clicks the "Status" link to access the stations dashboard and clicks on the "Create network" button. Upon supplying the relevant information, a network is created, and the user can proceed to click on the "Stations" link to create a new station or view and manage existing stations. Clicking on the "Add Station" button prompts for the basic information needed to create a new station under one of the networks associated with the current user. These setup operations are done once for each station or network.

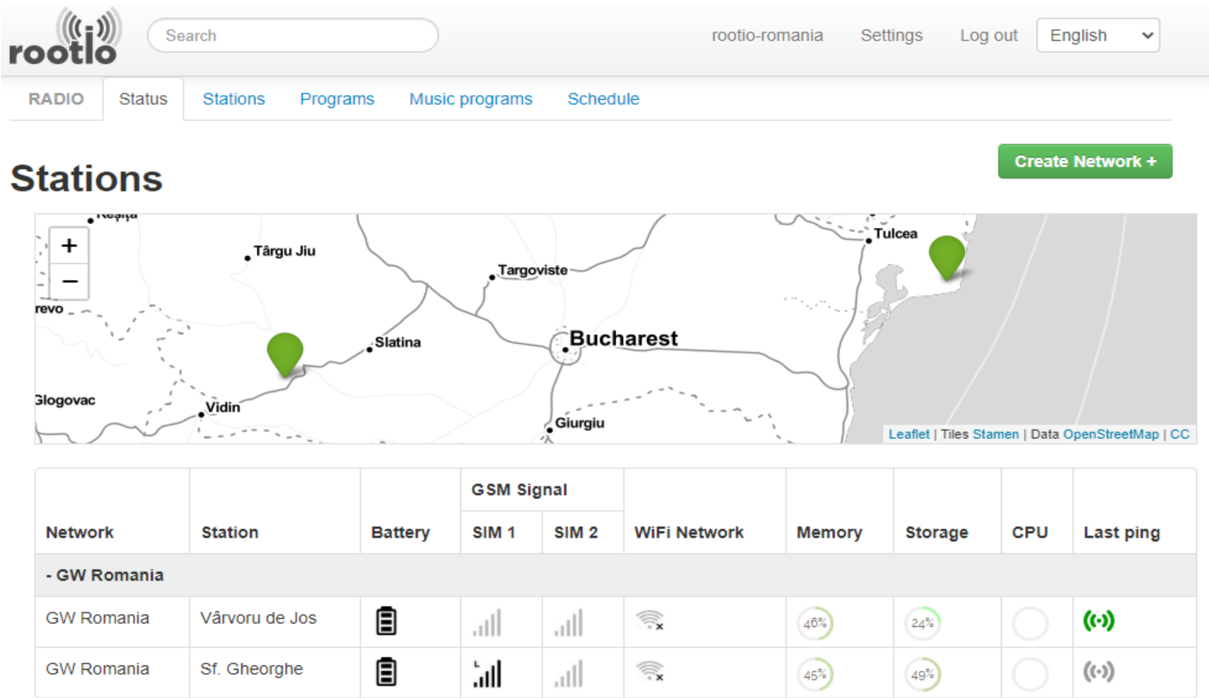



**Figure 3.8: The status page showing a network and status of its associated stations**

### 3.4.3 Sourcing content

The RootIO platform allows for the sourcing of content from various sources. The user can either upload their own content or rely on content being produced elsewhere. I also offer mechanisms for sourcing audio content en masse from within the community using telephony, a technology that is ubiquitous in many distanced communities. Below, I expound on the support available for sourcing content from various sources.

<u>Pre-recorded audio media:</u> The platform supports upload of media in two common formats: MP3 and wave audio. To add pre-recorded audio files to the content repository, the user navigates to the content management section by clicking "Manage content" in the main menu. The user then clicks on the "Tracks" link to view existing tracks and can click the "Add" button to add a new track, which is the term I use to refer to a collection of audio files. Upon supplying the name and, optionally, a description, the track is created and is visible in the listing of available tracks. Upon clicking on the "files" link available for each track, the user can see all previously uploaded files, as well as the option to upload new files to a particular track.

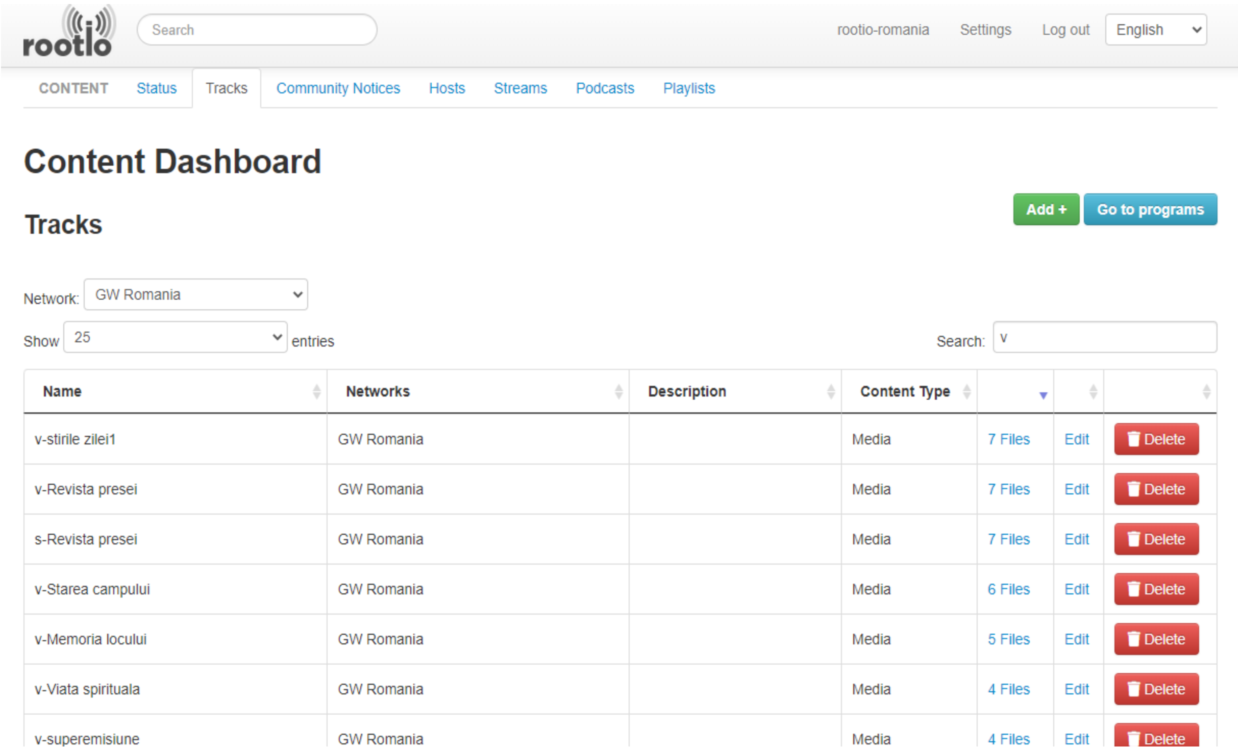



**Figure 3.9: A listing of media tracks showing available files per track**

When the files are added to a particular track, the user also has the option to define whether they want all the files played in a single instance of a program (for example, a news program may play an opening jingle, followed by the news file and an exit jingle), or as a single file per instance of a program; for example, in content that is serialized in episodes, such as audio books. This is done by manipulating the toggle switch “play episodes” on the files listing page.

Design decisions still need to be taken on some edge cases, such as when more slots are included on the schedule than there are files in a track. In some cases, such as audio books, it makes sense to play the last file for as long as no new file is uploaded. In other cases, such as informational programs, it makes sense to cycle back to the first files and loop through content, but, in this case, it is not clear how to resume programing once new files are added. In deployments, I pointed this out to the teams, who made sure there was always new content, and that old content was removed when it was no longer relevant, to avoid the risk of it playing again.

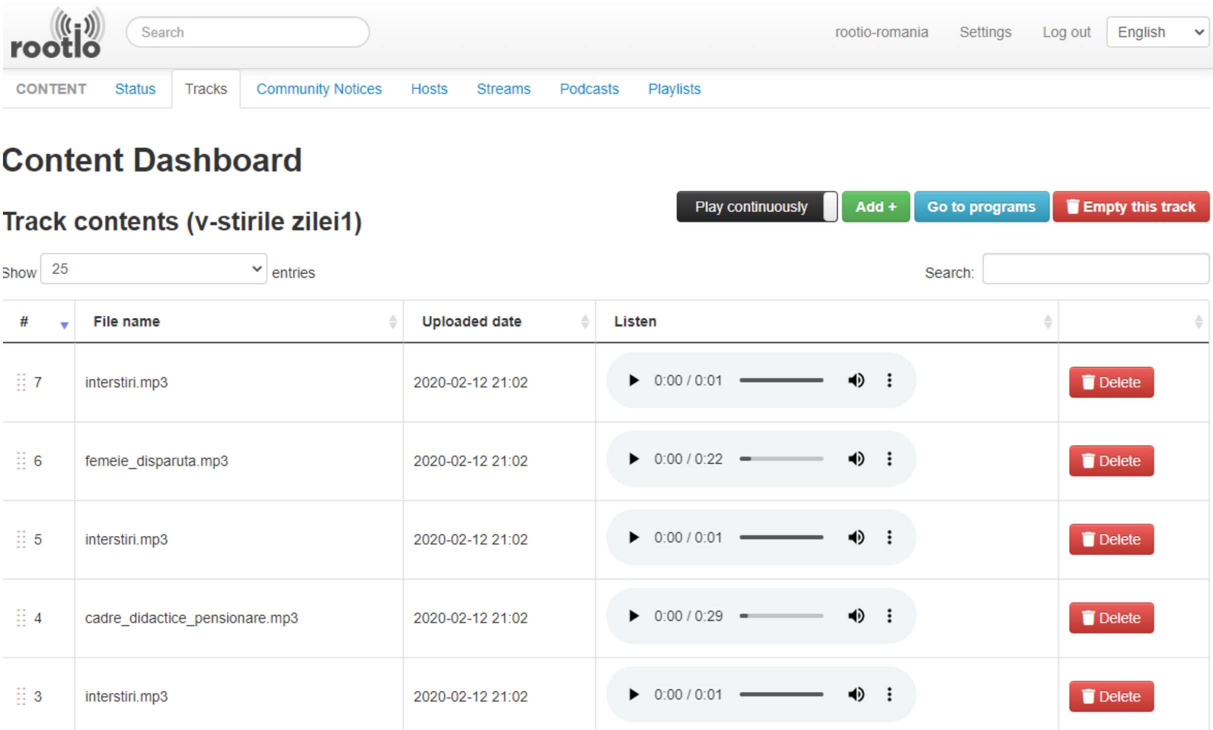



**Figure 3.10: A listing of files uploaded to a track**

Streams: To define a stream as a source of content, a user will click on the “streams” link in the “Manage content” section. This lists all existing streams and allows for the creation of new streams. The user clicks on the “Add” button to define a new stream source. After supplying the name and URL, and an optional description of the stream, the stream is created and is available as a source of content that can be incorporated into programs.

All the streams defined are visualized with an embedded media player that allows users of the platform to listen to the stream, in a bid to determine in advance whether the stream is playable by our scheduling system. My observation was that many communities tended to define links to pages that had embedded media players and JavaScript functions that played media when the page was loaded. This was because, when they clicked on the page, they heard audio on their devices but did not realize that the system in the background requires a link to actual audio content. To avoid this potential mix-up, the player allows a user to know if the stream they defined is playable or not by the RootIO platform.

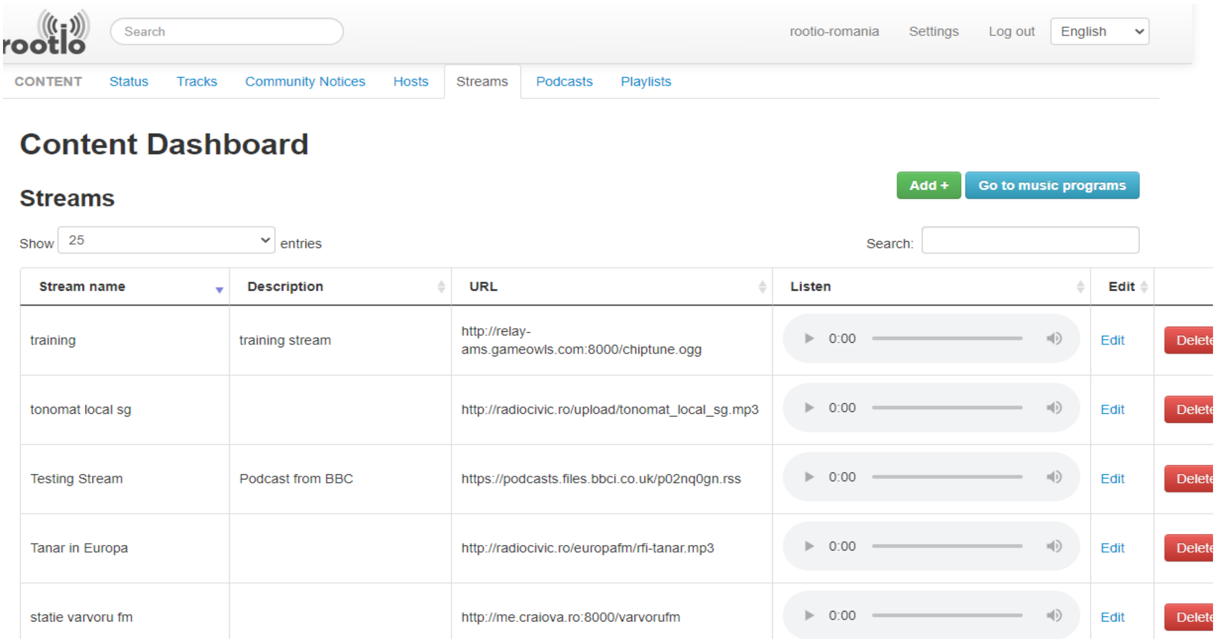



**Figure 3.11: A listing of streams available for inclusion in programs.**

<u>Podcasts:</u> To source content from a podcast, the user will click on the "podcasts" link and access a list of all previously defined podcast sources. The user then clicks on the "Add" button to define a new podcast source. After supplying the name, the URL for the RSS feed of the podcast and optionally a definition, the user can save the podcast source and will see it among the available podcast sources.

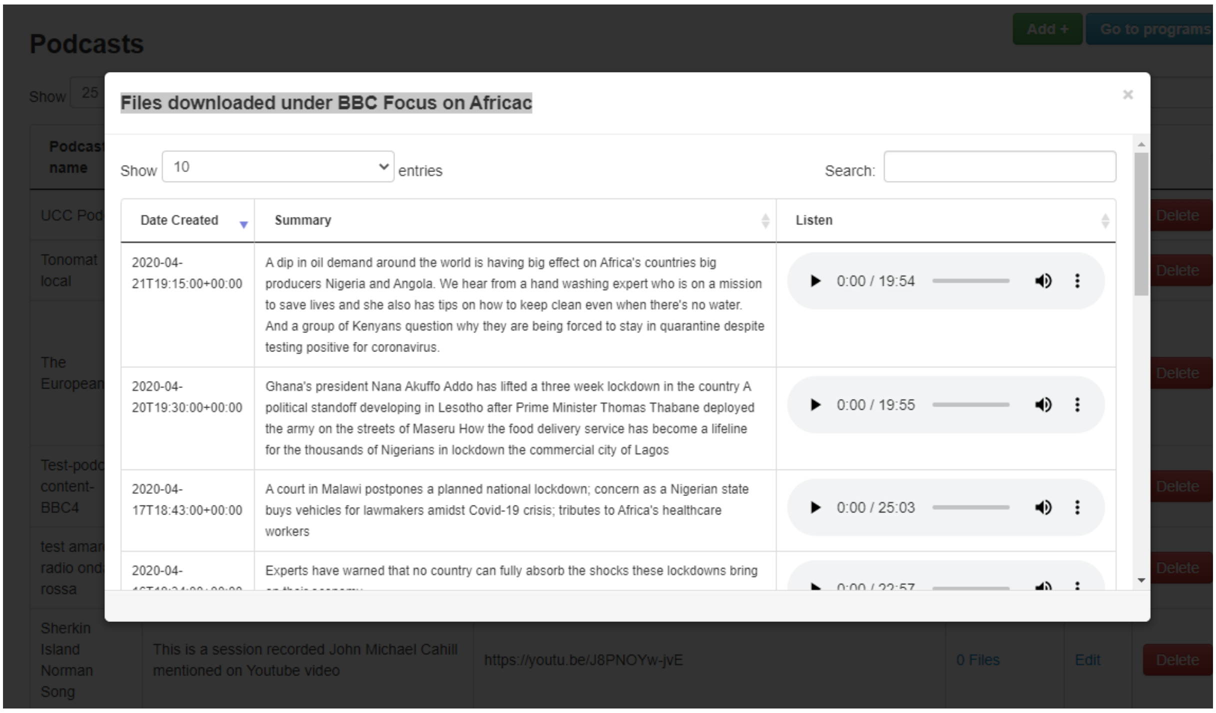



**Figure 3.12: A popup showing files syndicated from a podcast source**

In the background, the podcasts service will then download files for any new podcasts published on the specified URL feed. Clicking on the "files" link on individual podcast sources will show the files downloaded under that podcast. As is the case with streams, the listing features a media player that allows a user to preview the downloaded podcast, as well as get an idea of the duration of the audio file.

<u>Community voices:</u> This option allows for members of the community to easily use the station to pass messages within the community. By default, each station will have an IVR menu that is accessible via

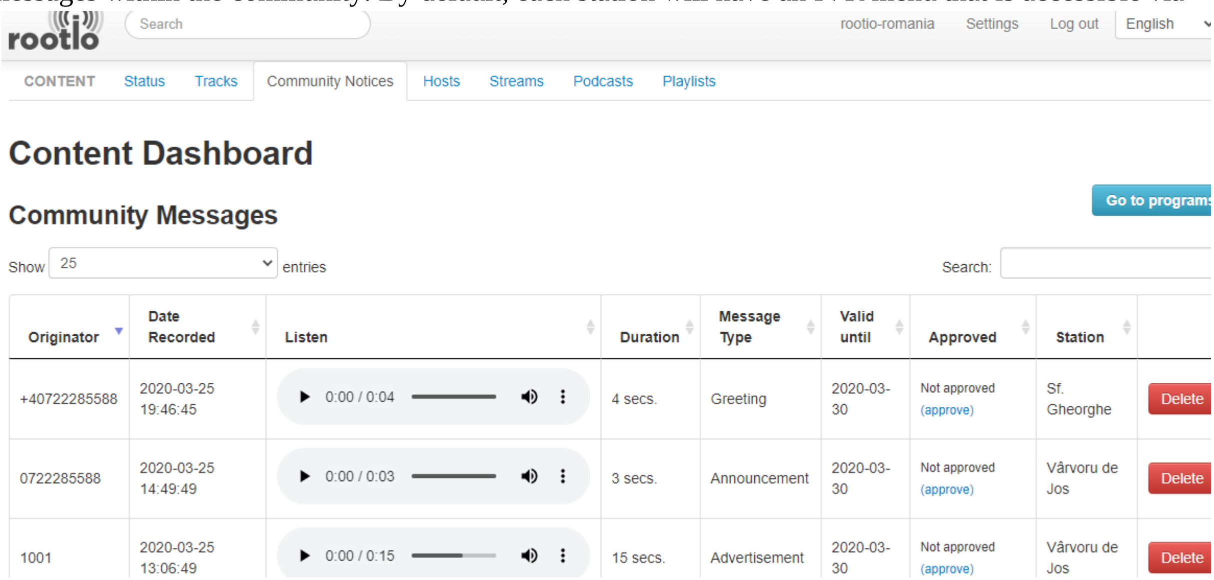



**Figure 3.13: A listing of messages originating from within the community**

telephony. By calling into a predefined number for each of the stations, a member of the community is taken through a menu that asks them what type of message they want to leave (advertisement, announcement, greeting), the number of days for which this message should play on the radio (1-9), and then are prompted to record their message and press the # key when done. Upon recording their message, the user is asked to press a digit to listen to their message, save it or discard it.

Upon saving the message, it is then listed among the content coming from the community and can only be aired once an administrator of a station approves it as fit for broadcast. In the event that most of the messages are appropriate, the approval process can be cumbersome and mostly unnecessary. A proposed enhancement here is to utilize a toggle option that chooses between playing only approved messages and playing all messages except those that are flagged. Alternatively, rules can be built in to automatically trust users that have a history of good use of the system while subjecting new users to the review process.

Station administrators have the option to customize the prompts that are played on each step of the menu and can rely on pre-recorded files, or harness the inbuilt TTS integration already built into the RootIO platform, to synthesize speech in up to 16 languages as of May 2021. By default, this menu is in English, but it is easy to adapt the system to generate prompts in the defined language of the station, if supported by the TTS provider that RootIO relies on.

<u>Playlists:</u> Station administrators can also create playlists based on the catalog of music that is available

on the phone located at the transmission site. As I will detail in chapter 3.5.4, the phone sends up to the cloud server a list of all the songs, albums, and artists available in the media library of the phone. By clicking on the "playlists" link a user can access existing playlists and click on the "add" button to create a new playlist. Upon supplying the name, description, and station for which this playlist is being created, the user can save the playlist and see it in the list of existing playlists.

To add an item (song, artist, album) to the playlist, the user clicks on the appropriate link on the record of that playlist in the listing of playlists. The user then has access to an overlay screen, with the artists/songs/albums available on the phone at the transmission site of the station for which the playlist is defined. The user then checks the items that they want included on the playlist, and this information is saved and later relayed to the phone at the transmission site. Below is an example of the information obtained from the REST API for listing playlists for a particular station. In this example, the station has one playlist containing two albums, three artists and eight songs.


```
{
      "playlists": [{
             "title": "Latino - Raspi 1",
             "albums": [{
                    "title": "Elegancia Tropical"
             }, {
                    "title": "Mundo Meu"
             }],
             "artists": [{
                    "title": "Gilles Peterson"
             }, {
                    "title": "Bomba Est\u00e9reo"
             }, {
                    "title": "Sanchez"
             }],
             "songs": [{
                    "title": "Is this the way"
             }, {
                    "title": "Blow Your Nose"
             }, {
                    "title": "Little heros"
             }, {
                    "title": "Higher High                    "
             }, {
                    "title": "Hold The Line"
             }, {
                    "title": "You know where to find me"
             }, {
                    "title": "My Father's Home"
             }, {
                    "title": "Cool down"
             }]
      }]
}
```


**Figure 3.14: A snippet of a playlist record with artist, album, and song info**

Playlists are different from other content sources in that they are created from content available on the phone and therefore form the basis for long-running programs that are delegated to the transmission site. These are referred to as "music programs" since they typically consist of music, though this could be any other audio content stored on the phone. Indeed, one challenge the team consistently faced was how and where to use existing language from radio – music player playlists, etc. Terms like "playlist" are already loaded with meaning for users, regardless of their specific uses in the sphere of radio. The project became "locked" into some of the early decisions, as in the case of "music programs."

Creating a host: A host is a member of the community, or knowledge expert, who conducts an interactive show on a RootIO station. The core concept is to allow for live discussions on the station facilitated by telephony in lieu of a studio setup. To define a host, a user clicks on the "Hosts" link to access the listing of already defined hosts. Upon clicking the "Add" button, the user is prompted to supply details of the host, with the name and phone number being the only mandatory fields. Upon clicking save, the new host is created and can be seen in the listing of existing hosts. In section 3.4.6, I explain how hosts facilitate talk-shows and how to add them to the schedule of the stations with which they are associated.

### 3.4.4 Creating a program

Programs are the building blocks of programming on RootIO stations. Content produced from various sources is aggregated into programs which are then laid out on the schedule. Programs therefore serve to conveniently group content for inclusion on the schedule. It therefore helps the reader to visualize the station schedule as a collection of programs, and a program as a collection of content from various sources. The platform makes distinctions in programs based on whether they are scheduled on the phone at the transmission site, or in the scheduler of the cloud platform.

Music programs: These programs are made up of music playlists and stream content. These programs play locally at the transmission site and therefore serve to relieve the load on the server as well as alleviate dependency on the Internet and telephony networks. This affords the transmission site autonomy to keep broadcasting content in the event that the server is not streaming content to the transmission site. These programs therefore serve as the background canvas, and as a recommendation cover the entire schedule, but can be interrupted by programs orchestrated by the cloud server. As such, these programs are limited to less dynamic content, such as playlists and streams, that can be stored on, or sourced by, the phone at the transmission site.

To create a music program, the user clicks the "Radio dashboard" link on the main menu, and then clicks on the "Music programs" link. The user then has access to all the existing music programs. To create a new music program, the user clicks on the "Add" button and is presented with a form that prompts them for the name, description of the program, and duration of the program. A basic board that acts as the content holder for the program, as well as boards with different playlists and streams for use as content sources, are also present. By dragging a stream or playlist onto the content board of the program, the user effectively defines that playlist or stream as the source of content for this program. Upon clicking the save button, the program is created and can be seen in the list of available programs.

Music programs allow for a mix of playlists and streams as content sources, streams, however, take precedence, and playlists are played if a particular stream is not accessible owing to a lack of Internet or other reasons.

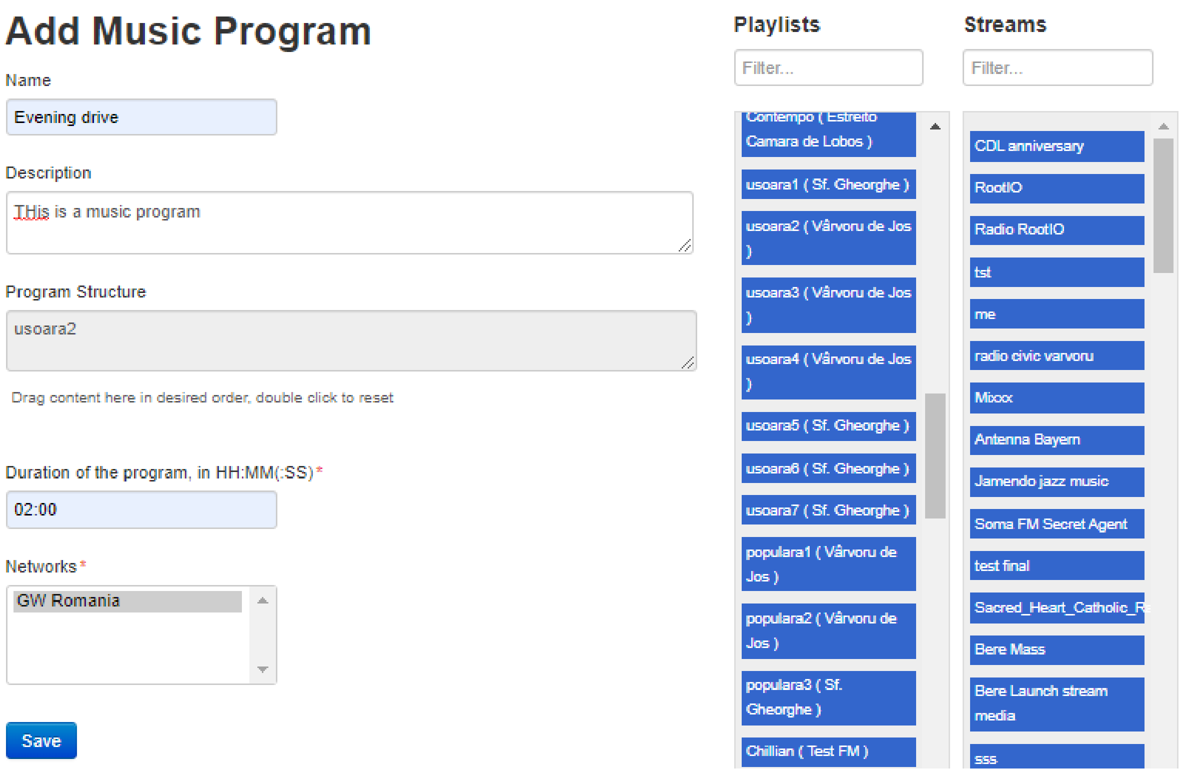



**Figure 3.15: Form for creation of a music program with content source options to the right**

<u>Programs:</u> These follow the original concept of programs streamed from the server to the transmission site and are therefore orchestrated by the cloud server. These programs are constructed from all the other content types except streams and playlists. While it is technically feasible to stream audio content to the phone via telephony channels (both PSTN and SIP), it makes little sense because of the high Internet usage, as well as the fact that telephony codecs are best designed for spoken content and would therefore degrade music content.

The process for creating a program is identical to the process of creating a music program. The user clicks on the "Programs" link and is presented with a listing of existing programs. On clicking the "Add" button, the user is presented with a form similar to that used when creating music programs, except in this case the available content sources include media tracks, podcasts, hosts and content from communities. Upon clicking the "save" button, the program is created and can be seen listed among the available programs.

A program can contain content from more than one source. In this case, the content from one source is played and when it is deemed to be finished, the content from the next source is played, until there are no more content sources. It is, however, not always easy to define what it means to play content from one source to the end. For instance, in the event that a program contains a host and an audio track, it is not easy to know when the host's talk show is done, given the flow explained in chapter 3.4.6. The users of the platform were therefore careful to mix content types whose flow is determinate, or to include non-deterministic content as the last item. A future improvement will be to implement intra-programming scheduling, allowing content within a specific program to have set time-bounds, as is the case with programs on the schedule.

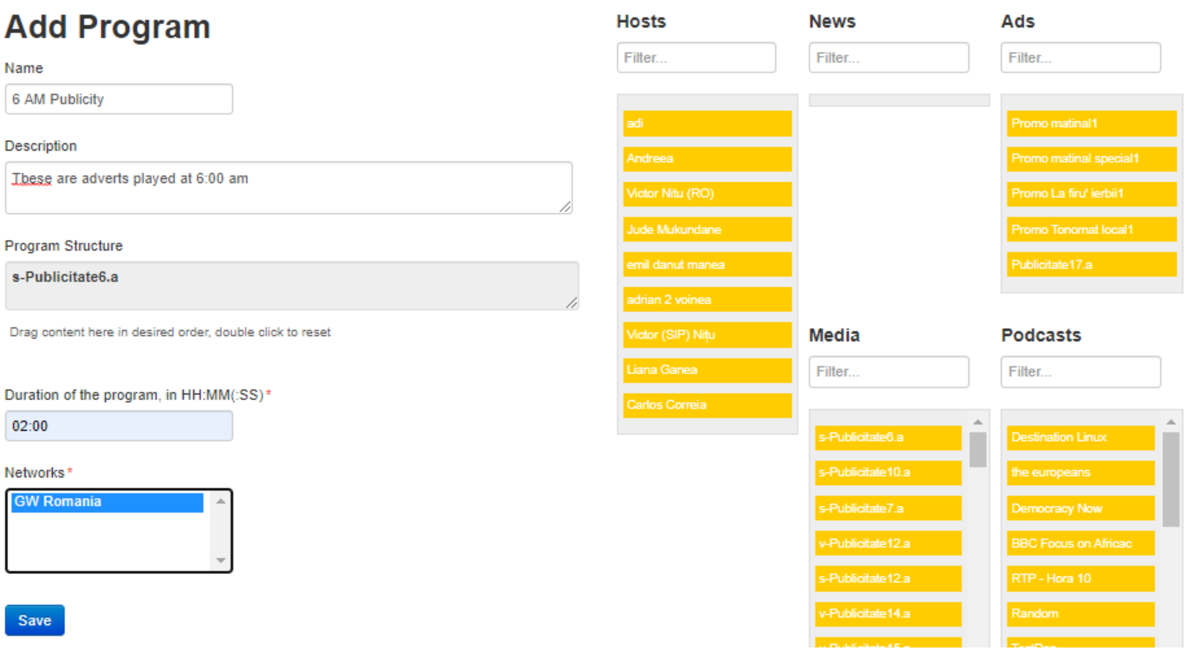



**Figure 3.16: Form for creation of a cloud program showing different content source options**

### 3.4.5 Scheduling programs

Scheduling programs is the last step in the pipeline of getting content to broadcast on RootIO stations. However, programs can be created, scheduled and content added to them later, but this flow involves a lot of data updates behind the scenes, both on the cloud platform and on the phone, and is therefore not recommended. The schedule instructs both the cloud server and transmission site how and when to play the various programs.

To access the schedule for a particular station, the user clicks the "schedule" link and is presented with a summary page showing the previous, next, and current programs on each of the stations within their networks. By clicking the "schedule" link on an individual station, the user has access to a calendar-style page showing the current schedule for that particular station. To the left of the schedule is a board that contains all programs that can be added to the schedule for that station.

To schedule a single program instance, the user drags a program from the board and drops it on the desired time slot. The scheduler also features, for convenience, a recurring scheduler that allows for the creation of a repeated schedule across weeks. The recurring scheduler allows for granular options such as the selection of particular days of the week, with the only limitation being that the broadcast time has to be the same for the selected days. Though the default view is weekly, the scheduler allows for daily and monthly views.

Music programs appear in blue to distinguish them from cloud programs, and it is recommended that these are scheduled for every available time slot in the schedule. The logic is that in the event that a cloud program cannot be initiated, the music programs will play on the device at the transmission site. As such, every time a music program is scheduled, moved, or deleted, a record is sent to the transmission site informing of the change in schedule.

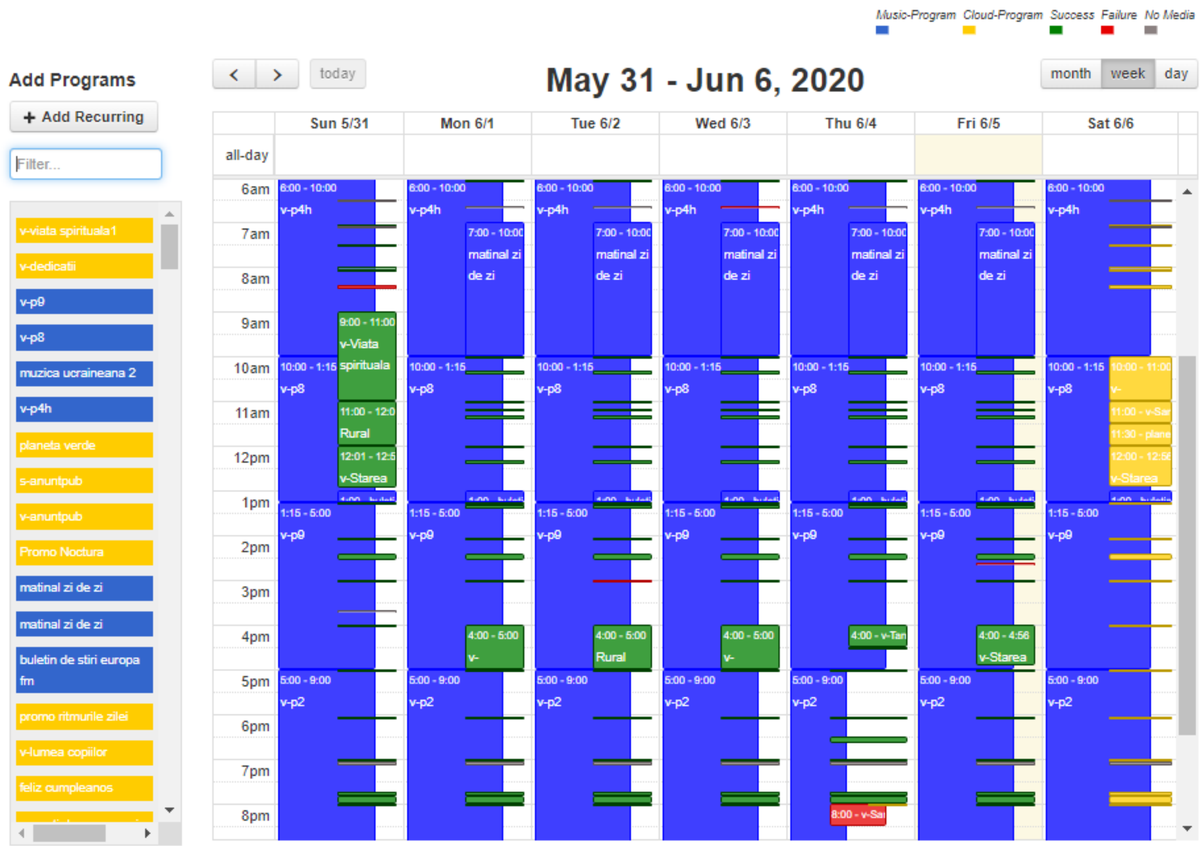



**Figure 3.17: A station schedule showing music programs (blue), successful cloud programs (green), failed cloud programs (red) programs without content (grey) and pending cloud programs (yellow)**

Cloud programs, on the other hand, run on the server and any changes made to them are communicated to the scheduler (based on Python APScheduler) service on the server via a messaging queue (RabbitMQ). Cloud based programs are yellow by default but change color to reflect the status of the program upon being processed by the server. Yellow programs are awaiting processing, while grey programs are those for which no content is defined, or the currently defined content source is empty. This might be the case if a media track is defined as the source, but no files are uploaded to the track, or if a podcast is defined as the source but no new files are published to the podcast.

Once a program is processed, its status changes to reflect either success (green in color) or failure (red in color). A program is deemed successful if the cloud server established contact with the transmission site and played the specified audio content to completion. In the event that contact is not made, is interrupted, or the specified content is not played to completion, the program is marked as failed. For each program, optionally, an email summary is sent to the administrator of the station detailing the steps that were taken for a particular instance of a program broadcast.

### 3.4.6 Airing content

The scheduler at the transmission site and the cloud-based scheduler run concurrently. Each scheduler is tasked with orchestrating its programs, but the two work together to deliver a seamless radio experience to listeners. At the transmission site, the scheduler, once started, also listens for incoming calls via both the phone's telephony system and the IP telephony system built into the RootIO app. If a call is received and matches one of the numbers previously authorized, the phone-based scheduler fades out any audio that is currently playing and accepts the incoming call. Once the cloud-based scheduler is notified of the acceptance of the call, the predefined audio is fed into the call and broadcast at the transmission site. Below we look at the models for airing programs with different types of content.

Podcasts, Media, and Community content: This content is streamed by the cloud server and is played through a telephone call on the transmission site. The cloud-server will first determine if there is any content to play before initiating a call to the transmission site. Given that a cloud-based program may contain content from various sources, each of these sources will be inspected, and if all of them do not contain any playable content, the program is forfeited and marked as failed.

The next step, after content evaluation, is to select a route through which to contact the transmission point. In the event that the station is classified as having high bandwidth (in station settings), the station will register directly with the PBX hosted by the cloud server. In this case, no external telephony systems are necessary, and a call is originated directly to the transmission site via SIP. If the transmission point does not have IP connectivity, the call is originated via a PSTN network as explained in section 3.3.1.

When the phone call to the station is received at the transmission site, the phone will fade out the audio that is playing and answer the incoming call. Once this is done, the cloud server receives a notification that the call has been answered and proceeds to play a media file into the call. When the file is played to completion, the scheduler will check if the program contains more files, in the absence of which the call to the station is terminated and the program logged as a success. At the transmission site, the notification of this termination instructs the local scheduler to resume its own media play.

Pertaining to the different categories, small differences exist as to how their media is organized by the scheduler. In the case of podcasts, only the most recent file is played, and, as long as it is the most recent file, it may be repeated. In the case of media tracks, the play follows the order of the files (facilitated by drag-drop functionality). If the track is configured to play episodes, only one file is played per program, and the pointer is increased to the next file for the next program. In case that the option is selected to play all files, then all files are played in every single instance of the program. This is the preferred option for scenarios such as news bulletins with an intro and outro.

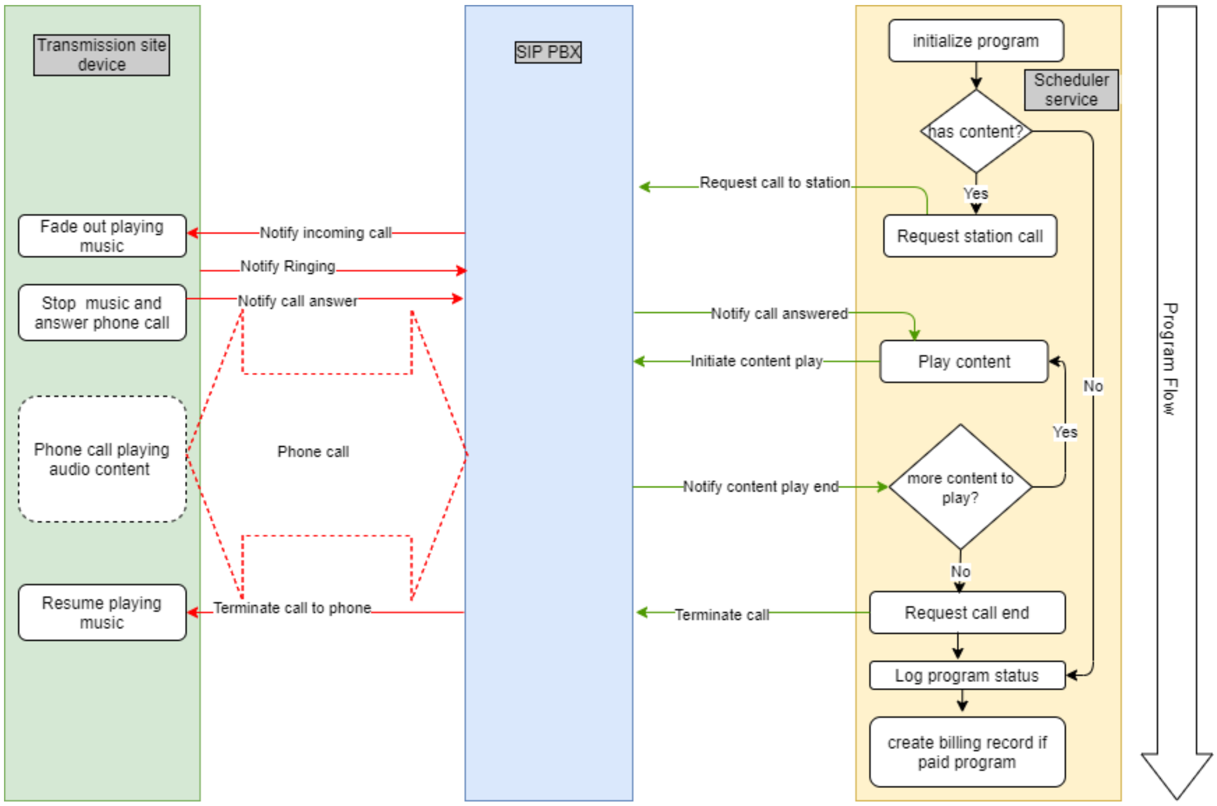



**Figure 3.18: Schematic demonstrating the interaction between the cloud server and transmission site to play content from the cloud**

For community content, the content is played for the number of days indicated by the creator upon recording. For each program instance, all existing files are played back-to-back and the whole process is looped for a configurable number of times.

Hosts: Users defined as hosts facilitate talk shows on the stations they are associated with. They can be included in a program definition, just like any other content source, but the flow of their programs is entirely different. At the scheduled starting time of a talk show program, a host receives a phone call from the cloud server. They are prompted to press a digit if they are ready to take the call, or press a different digit if they are not ready.

When the host confirms that they are ready to host the talk show, the cloud server calls the transmission site via either SIP or PSTN. Once the call is established, the host is informed that they are live on air and can begin speaking, with their call being broadcast to the community. The host has access to a menu that allows them to perform certain actions by pressing a particular number during the call.

In essence, the talk show is a conference call with the device at the transmission site acting as one of the participants. With this in mind, it is possible to add participants to the call using different modes.

To begin with, the host can press the “3” key to turn on auto-answering; in this mode all callers into the conference call are automatically bridged in. The host simply needs to press the “3” key again to turn this functionality off. In either case, the host will get an audio message, audible only to them, informing them whether the functionality has been turned on or off. By pressing the “4” key, the host puts the talk-show in queuing mode.

| **Key** | **Function** |
|---|---|
| 1 | During call setup, pressing 1 means the host is ready and the talk show commences. During the talk show, this puts the station in “wake mode”. |
| 2 | During call setup, pressing 2 means the host is not ready, so the talk show setup is aborted. |
| 3 | Toggles auto answer on/off. When on, all calls into the conference are automatically answered and bridged into the call, otherwise rejected. |
| 4 | Toggles queuing on/off. When on, all call attempts are recorded and put into a queue so the host can dequeue numbers from the call and call them back. |
| 5 | Pressing 5 dequeues a previously queued number and originates an outbound call. Once answered, the call is bridged into the conference. |
| 6 | Terminates a call that was previously bridged into the conference either by automatic answering, calling back from the queue. |
| 7 | Terminates a call that was originated by the host entering a number to call (by pressing the * key explained below), |
| 9 | Allows the host to take a 5-minute break. The station calls them back after their break and the setup process is repeated. |
| * | Pressing * the first time allows the host to enter a number and press # to originate a call to that number. Pressing * again before pressing the # key clears the buffer and numbers can be used to access menu options. |
| # | Upon pressing the * key, the host enters a number and pressing # results in that number being called and bridged into the conference call for the talk show. |

**Table 3.3: Menu functions available during talk-shows and associated keys to press**

In the queuing mode all calls into the conference are rejected, but the number of the caller is placed in a queue. The host can then press the “5” key to dequeue a number and originate a call to its owner. This allows for free participation in talk-shows within the community at a cost of roughly 0.25€ per day to the station, in places where calling is predominantly pay-as-you-go and people rarely have credit on their phones. A proposed UX improvement in this case was to answer the call and inform the caller that they will be called back, given that the current rejected call is easily taken for a failed call, which results in retries. This however might incur an expense on behalf of the user, especially in pay-as-you go markets. A better approach would be to send back an SMS providing the same information.

For callers who are in the conference call, whether by auto-answering or call-back, the host can press the “6” key to terminate their call. This gives the host control over any unruly participants and time management. The host can also bring in another participant by dialing their number. This is achieved by pressing the “*” key, followed by the # key, the number of the participant to be called and the “#” key again. Even with an “invited” participant, it is possible to accept calls from the larger community. The “7” key therefore helps to terminate calls to “invited” participants.

Finally, the host can press the “9” key to take a five-minute break. The station hangs up on them and calls them back five minutes later. It is easy to make this period configurable based on the host, station or even program. Should the host merely hang up, the station will not call them back, but rather the host can call the station, which will reject their call and instead call them back. This is so that the host does not incur expenses while conducting the show. It is worth noting that in this scenario the host is called back only if their program is still running on the schedule.

An alternative implementation demanded by communities in Romania allows the host to press the 1 key during the course of the talk show, putting the station in “wake mode”. In this mode, participants can call into the station and subsequently a call is originated to the host of the scheduled program. The host then interacts with the participant behind the scenes and, if deemed necessary, the host can press a key to bridge their call to the transmission site, effectively putting the conversation on air. This allows for the transmission site to play music which will then be interrupted by a host when someone wants to go on air.

Hosting talk-shows in this manner is a far cry from the traditional setting in which the host and guest are in a studio and interact face to face. The studio setting is a prevailing norm, which people are reluctant to move away from. However, practitioners building on the RootIO stack in Ireland were appreciative of the afforded cooperation with members based off of the islands on which stations were deployed [106]. Also, recent events such as the COVID-19 pandemic have rapidly ushered in remote working as a potential new normal [10,30]. Broadcast practitioners have also needed to innovate to work from home in a model [89] very similar to what RootIO proposes. I believe that the RootIO technology for distributed talk-shows will receive more appreciation, similar to that which is shown for tools such as Skype® and Zoom®, especially now when physical distancing has been imposed and age-old norms such as physical offices have become optional overnight.

## 3.5 Functions facilitated by the RootIO mobile app

The RootIO Android app sits on the phone at the transmission site and executes functions as delegated by the cloud platform. As previously mentioned, the core function of the app is to provide autonomy at the transmission site given that it is not practical for the cloud server to constantly stream content to the transmission site. To this end, the RootIO app carries out some functions and routinely communicates with the cloud platform to synchronize the status of the transmission site with the cloud platform as well as pulling down new logic for playing content at the transmission site.

The RootIO application is divided into 6 modules that run as background services launched by the app thus:

- Radio Service: This plays audio content, both local and streamed, as defined by a program schedule.

- Telephony Service: This manages calls coming in via PSTN networks. It authenticates users, and answers or rejects calls according to defined lists, all while notifying other app components of changes in call status. This was later merged into the Radio service, since sometimes messages to the Radio service were not delivered, resulting in media playing over phone calls.

- Diagnostics service: This service periodically collects metrics about the phone, regarding the performance and usage of networks, memory, CPU, battery and storage.

- Synchronization service: This is responsible for sending information about phone metrics, network information, call and SMS records, and the music library to the cloud platform. It also downloads program schedules, playlist definitions, calling access lists, information about the station to which the phone belongs, and information about frequencies for gathering/synchronizing diagnostics with the cloud platform.

- LinSIP service: This is responsible for SIP telephony using the LinSIP library. It is so named to differentiate it from the previous SIP service, which was implemented using the Android SDK's native SIP stack. This too was later incorporated into the Radio service for the same reasons as the Telephony service.

- Message service: This is responsible for intercepting incoming SMS messages, inspecting them for commands to the RootIO app, and executing any associated actions. Given the importance of this service, it was later implemented as a standalone part of the app that can be called by the SMS system even when the RootIO app is not running.

Below we delve into the functions supported at the transmission site courtesy of the RootIO app and its constituent services.

### 3.5.1 Managing phone calls

One of the major reasons for choosing the phone as the platform for the transmission site is the solid integration of telephony. For the RootIO setup to perform ideally, the app needs to manage phone calls, that is, permit desirable calls while declining unwelcome calls. While previous versions of Android made use of hacks to answer and terminate calls, these were blocked in versions from Jellybean to Nougat. Android 8 (Oreo) however allows for programmatic answering of calls but not termination.

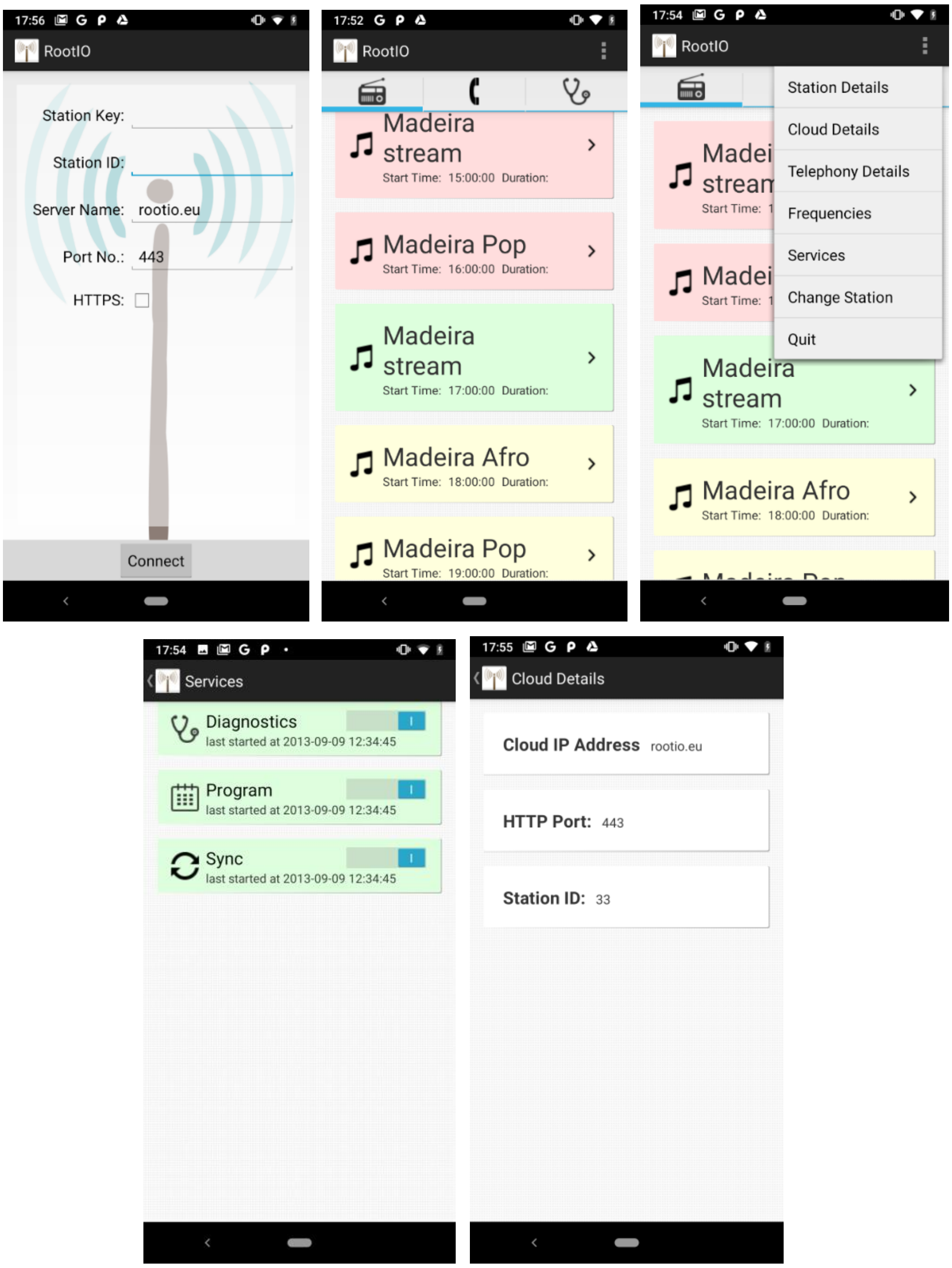



**Figure 3.19: Screenshots from the mobile app showing the screen to configure the cloud server (1), programs schedule (2), options menu (3), screen to start and stop the services run by the phone (4) and instance to which the phone is currently connected (5)**

To manage calls via the SIM card, the app depends on an access list of allowed numbers supplied by one of the API endpoints supplied by the cloud platform. If a call comes in, this list is checked and if the number is on the list, any audio currently playing is faded out and the call is answered. In the event that the number is not on the whitelist, the call is left to ring given that there is no way to programmatically terminate calls. Because of this, the phone is set to “do not disturb” mode such that the phone does not ring, even through the headphone jack whose audio is being fed into the FM transmitter. Periodically, the phone sends up a list of all the calls it received and declined, and these records can be visualized in the web platform.

For SIP calls, the rules are slightly different. The app accepts calls only from the PBX to which it is registered as a client; as such the filtering is only on the address of the originating server. Since control of the SIP server is under the administrator of the RootIO instance, this should be enough security, though care needs to be taken to make sure that there are no rules that allow for the forwarding of rogue SIP traffic from the PBX to the phone as a result of SIP spoofing [34].

### 3.5.2 Taking diagnostics

Given that the design is for transmission points to be in relatively hard-to-access points, it is important to have an idea of the performance of various components of the phone without having to physically access the phone. The app periodically collects vital indicators about the phone, and these are sent up to the cloud to show the status of the phone. These indicators include:

- Memory usage
- Storage utilization
- CPU utilization
- Connectivity to WIFI network
- Connectivity to phone networks (including type and strength)
- Battery status (percentage of charge)
- GPS Location

These diagnostics are taken at a configurable interval, with the default setting being once every 30 seconds. Increasing the granularity results in increased data to store and sync to the cloud platform. They are included in the information that is periodically sent to the cloud server.

### 3.5.3 Persisting program and playlist information

Periodically, the app on the phone will contact the cloud server for information pertaining to playlists and programs. In either case, the phone contacts the relevant API endpoints, querying for changes to playlists or programs since a supplied date, which is normally the date of the last changes received by the phone. If the phone is newly setup and no records exist, it will request all new changes affecting programs in the future. The app can also delete, add or update scheduled program items to match the current layout in the master schedule maintained by the cloud server. In addition, the phone also receives a list of all playlists defined in the cloud for content currently stored on the phone. Both these interactions return data in JSON format, which is parsed and stored in an SQLite Database for easier querying during program execution.

### 3.5.4 Synchronizing media content

The content from which playlists are created resides on the phone and is organized periodically by the Android system into its media library. The app queries this content and sends it up to the cloud in a format organized hierarchically by artist, album, and individual songs. This allows for the creation of playlists based on artists, albums, and individual songs; playlists are later synchronized to the phone as described in section 3.5.3, and can be referenced in programs that play locally at the transmission site. Below is a sample JSON representation of content being synchronized to the cloud.

### 3.5.5 Playing music programs

Music programs, as previously mentioned, play locally on the phone and are orchestrated by the RootIO app. The app gets a schedule of programs from the cloud server, as well as the associated playlists or streams to play for each program. The app schedules programs according to the schedule received from the cloud and when the time for a program arrives, the scheduler looks at the associated playlists or streams and assembles the content into a master playlist that is fed into the media player.

Regarding playlists, the scheduler will build a catalog of songs, artists, and albums included and thereby build a superset of content to be played in the program slot. When artists and albums are defined in a playlist, the media set is expanded to include all songs that belong to the specified artists and albums. Iteratively, a media file obtained from the set is passed to a media player (based on Exoplayer), and upon its completion, the media player notifies the program service to pass the next file. In the event that the content runs out before the end of the program, the set is replayed after shuffling the songs.

For streams, the URL is simply passed to the media player, which plays the stream or keeps trying to start it if initial attempts fail due to Internet outage. If a stream is specified along with a playlist for a particular program, the playlist begins if the stream cannot be played, and the stream is retried after the conclusion of every song in the playlist. There are potential design alternatives here, such as not reattempting the stream since the listener experience may be degraded if it is unreliable and multiple reattempts are made. Alternatively, for critical real-time informational streams, it may be necessary to try as soon as possible, rather than wait for a song that may last for many minutes.

Two types of playlists are given special consideration and conveniently add much needed features to the station: A playlist with the word “random” in the title instructs the scheduler to play random audio content from phone’s media library, while content specified in a playlist named “jingles” is overlaid over any other content playing at the transmission site. In the latter case, the volume of the content currently playing is reduced by 80% and audio content on the “jingles” playlist is played at full volume, creating an overlay effect. This is used for jingles such as those informing the callsign of the station.

## 3.6 The Different users in the RootIO ecosystem

RootIO straddles the border between broadcast media and online platforms. This design embodies the core philosophy of RootIO that proposes situating modern digital technologies behind traditional broadcast and basic voice and SMS telephony offerings by telecommunications companies. Whether on the digital or analog side, RootIO has users that interact with stations at different levels. The RootIO model therefore involves users interacting with radio stations through different channels; some managing the station via the web portal, others delivering information via mobile phones, and others simply listening to the station.

We treat all these users as part of the same ecosystem, and each RootIO station as an information platform similar to online platforms such as those that facilitate social media. The goal here is to clearly delineate these users such that, when features are being proposed, the target users and their interaction mode with the station are considered. It is also important that the overall interests served by the station are shared between all user categories. I observed communities in many scenarios that relied on users, especially for station administration, who were working in the community, but did not reside within them. In Romania, for example, MedAlert was based in Bucharest and did most of the station administration, while in Northern Uganda, most of the administration was performed by the Northern Uganda Media Club which is based outside the communities of deployment but focuses on the region. This model aligns with previously cited research that legitimizes the role of intermediary users in ICT4D settings.

Each of these user groups has their motivation for using the radio station and the user categories are not mutually exclusive. For some, it is the need to disseminate information about their activities, for others it is their business to curate information and sell it; some are motivated by their interest in advertising their products and services, while others are seeking information and entertainment. Below we delve into these user scenarios and specify which are concretized in the RootIO platform and the measures that are taken to represent those that never interact directly with the web platform.

### 3.6.1 Administrator

The administrator role is global on a RootIO instance. The holder of this role has a view of all the programs, stations, content, and configuration information for the entire RootIO instance. This role allows for a user to support stations where there is no capacity to administer a RootIO station/network. The administrator can perform all the functions of all other roles, across all networks. Additionally, the administrator can add new records to category dictionaries such as language and gender choices. This role is reserved for members of the company or entity that manages the RootIO instance.

### 3.6.2 Network Administrator

The network administrator is similar to the global administrator role except that it is granted on a network (group of stations) level. The administrator of a network can:

- Configure station parameters and details.
- Manage content sources and uploaded/syndicated content.
- Invite, disassociate, and grant roles to users within the networks for which they are an administrator.
- Create programs and schedule them on stations within their networks.
- Archive content, content sources and programs that are no longer being scheduled on the stations within their networks.
- Review and approve messages left via the telephony menus for stations within the network.

The performance of duties associated with this role is possible only through the web portal. From my observation, the holder of this role belongs to the entity that is responsible for ensuring compliance with broadcast content regulations and bearing experience in the broadcast media industry.

### 3.6.3 Network users

Network users can carry out activities on behalf of stations but cannot change critical settings relating to the stations that they administer. A station user, for example, cannot archive content and programs as is possible for the network admin, nor can they administer users within their networks. The idea behind network users is that RootIO stations feature routine core functions to keep stations operational. Defining programs, and scheduling/unscheduling them, are routine operations that run low risk of breaking regulations common in the broadcast space. It was observed however that in some contexts, like Romania, the broadcast schedule had to be filed with the regulator and adhered to. In this case, it might make sense to further restrict the ability of network users to schedule.

On the other hand, actions such as sourcing new content, or changing the frequency of the station (this is not yet implemented but is feasible), open the station up to potential copyright infringement and frequency encroachment. These functions are therefore not available to users whose main preoccupation is maintaining the programming on the station. This too is a role that is performed via the web portal and, as I observed, these were always based within the community. It was observed in Romania that a program schedule had to be filed with the broadcast regulator and adhered to, meaning the manipulation of the schedule may need increased restriction. This suggests the need for granular function permissions rather than permissions based on roles.

### 3.6.4 Content administrators

The content administrator role was never used in the time that I studied the use of the RootIO platform but would be required if the platform evolves in the anticipated direction. The content administrator role is designed to enable curators of content to source content and make it available to the platform for inclusion in programs. The idea is to enable independent media practitioners to source and utilize content in the platform following consideration of pricing agreements and copyright issues. The envisaged approach is that content administrators "publish" content in the platform; content which is advertised to different stations. In return, the publishers see how many stations subscribed to their content and how much they earned, or are supposed to pay, for their content that was played.

This allows network administrators and users to add content to their schedules knowing how much they would earn from it or pay for it. Additionally, the network users and administrators would know that they are insured from any copyright issues regarding the content they obtain from these content providers. This is a promising direction that could lead to a bridging between modern digital advertising and FM radio, as elaborated in chapter 7.2.5. This role is also entirely web based.

### 3.6.5 Hosts

Hosts are users who present topical discussions on RootIO stations, with the option to include other members of the community using telephony. In the traditional RootIO setting, hosts only interact with the station via voice telephony, even though they are defined in the web portal. Given the dependence on telephony, hosts may or may not be based within the community served by the station.

Where Internet allows, hosts may not be defined in the platform but can speak into a microphone, the output of which is fed into a stream chosen by the station. Given the history of FM radio, many hosts were uncomfortable with the idea of speaking into a telephone with no response from the callee. I observed instances of this in many communities and attempts were made to instate a studio-like setting. In Northern Uganda, the community in Patongo introduced a mixing board into the set up that allowed the use of a microphone, while in Romania communities relied on streaming. In the latter setup, a host talks into a microphone connected to a laptop and the output is channeled into a stream chosen by the phone at the transmission site. This means that talk shows are scheduled as "music programs" with a stream source, as detailed in chapter 3.5.5. Hosts can therefore take on a fluid role beyond that defined in the platform.

### 3.6.6 Community callers

Members of the community can call into an IVR menu that allows them to leave a voice message that is later aired on their station. Users of the station in this sense interact with the station using their mobile phones for voice calls. Besides calling into the station to leave announcements and notices using the self-service menu, these users can also participate in talk-shows if the host activates the options that allow them to call in or request call backs from the station in order to participate in talk show programs, as described in chapter 3.4.6.

Though these users are tracked in the system by their phone numbers, other data can be added to their records to subsequently enrich their profile and visualize them in the system as users with accessibility beyond what is possible if they only provide a phone number. Theoretically, decisions can be made based on features such as, for example, gender and age, in order to prioritize the participation of one demographic in scenarios of contention such as when calling into a talk show. It is also possible to build systems for dissemination of information such as sending text messages of news summaries or callouts for advertisements and news snippets. The inclusion, and representation, of these users in the platform is prime research and perhaps the most direct projection of the RootIO idea.

### 3.6.7 Advertisers and announcers

Not to be confused with the above-mentioned users, advertisers and announcers use the station to pass on pre-recorded adverts and announcements. These users already exist in mainstream radio and are mostly established corporations and businesses advertising their products and services. In my observation of RootIO stations, these tended to be established businesses as opposed to individual traders. These users pass their messages on to a volunteer from the station, who then uploads it into the relevant media track for subsequent airing on the station. Even though I provided self-service call-in menus that allowed members of the community to advertise their goods and services, I observed a category of entities that preferred the option of produced advertisements.

### 3.6.8 Listeners

Listeners are without doubt the largest user category of a RootIO station, and typically they receive the broadcast through their FM radio receivers. Given that FM radio is entirely one-way, it is almost implausible to pinpoint these users. The station in Patongo gave unique numbers to listeners every time they called into the station and managed to go a little over 100. Because listeners have no direct representation in the RootIO platform, it is important to put in place measures that ensure that they have a bearing on the operation and broadcast of the station. The failure of many community FM radio setups to prioritize listeners is what created the space that is occupied by RootIO in the first place.

In the early deployments in Northern Uganda, I realized that a small cohort of individuals close to the organizations hosting the stations also held control of the stations. I emphasized the need for a balanced management board selected from various demographics of the adult population of the community served by the station. Besides giving telephony access that allows community members to have their say on matters, including the operation of stations, I also included a section for the documentation of community meetings around the stations in the design. This helps keep a clear log of decision-making around the station. The log of meetings can be audited by a media organization to make sure that plurality and inclusion are upheld in managing the station.

## 3.7 How RootIO changes the community radio game

As mentioned earlier, FM radio for distanced communities is not a new venture. Community radio initiatives have been attempted, notably in Mozambique which boasts of 52 recognized community radio stations [133]. Visits to 3 stations in Mozambique did not yield examples where deploying as a community FM radio station has resulted in increased grassroots control, made it easier for interested communities to setup and operate their own radio stations, or increased the sustainability of FM radio in distanced communities. RootIO, on the other hand, introduces technologies and approaches that stop the encumbrances faced by typical FM radio stations from percolating down to setups owned by grassroots communities. Below we explore these features in detail.

### 3.7.1 Reduced CAPEX and OPEX

Cost remains one of the factors preventing the adoption of ICTs in grassroots communities. Given that these communities are most likely at the lower end of the economic scale, the resources available for investment in different ICTs are scant. As such, for RootIO to be plausible, it was important that the design addressed the expenses incurred in setting up the station (capital expenditure or CAPEX for short) and running it (operational expenditure or OPEX for short).

Typical FM radio stations are designed to cover large geographical areas, and this has huge implications on the cost of setup. To begin with, these stations must choose the best location to cover a certain region, usually spanning a radius of hundreds of kilometers. This often involves locating the transmission point on hill-tops where new construction must be undertaken for the housing of the transmission point and the tower on which to mount the antenna. By covering small areas, RootIO facilitates the colocation of stations within the served communities, thereby eliminating this cost.

The traditional model of FM radio involves the use of a studio. These vary in cost and sophistication but are a significant part of the station setup cost. RootIO does not offer a studio and therefore eliminates this cost while keeping track of the lost functionality for alternative implementation. The inclusion of streaming allows for the coupling of any studio to a RootIO station for only the time when this is needed, as opposed to having a permanent studio. RootIO also provides mechanisms for subscribing to content produced in high quality studio settings, such as podcasts from media houses. As such, the functionality of studios is not lost, but rather RootIO allows for FM radio where studios are not tenable.

There is a huge reduction overall in the hardware that is necessary to achieve broadcast. The transmitters used in the RootIO setup cost less than 500€ in almost all cases, the smartphones cost less than 300€, but the preferred Raspberry Pi computer costs 50€, and the antenna, cable, and tower all fall within an additional 1000€, making this one of the cheapest hardware stacks for operating an FM broadcast. The reduced hardware also results in reduced installation labor costs, as it is possible for two people to install RootIO stations in a single day.

Energy consumption is another area where RootIO stations eliminate a constraint. The transmitters are the heaviest consumers of energy, but at a 60W power rating, these too are infinitesimal in comparison to the transmitters used in typical FM radio stations. The investment in solar energy, though posing a huge capital expenditure, ensures that RootIO stations have a free energy supply, one of the biggest costs according to community radio stations that I interacted with in Mozambique and Uganda. Where the capital expenditure for investment in solar is non-existent, but grid energy exists, the RootIO ensemble draws much less power compared to the setups of community radios, making its operation much more tenable, even without solar energy.

By offering options that allow for automated and collaborative content sourcing, RootIO reduces the labor requirement that burdens community FM radio setups. These savings reduce financial pressure, which is always catalyst for stations expanding to earn more from advertisement revenue. This results in increased sustainability for stations with small audiences as proposed by RootIO. As will be demonstrated in chapter 4.4.5, in one of the remotest deployments, one of the stations demonstrates a healthy return on investment from the operation of the station.

### 3.7.2 Telephony for increased input and talk shows.

Another proposition of RootIO is increased input and participation in FM radio by people that depend on it. RootIO harnesses telephony to circumvent two challenges: the lack of a studio and censoring of community voices. As emphasized in the name, RootIO seeks to increase the input and output of content originating from within grassroots communities. The telephony technologies that I present to users are familiar and require no increase in literacy. In so doing, RootIO allows these communities to do more with technologies that they are already familiar with.

The lack of a studio affects talk shows more than any other program format. In all the community radio stations that I visited, there were studios furnished to different extents with the common factor being the ability to accommodate two or more people talking individual or shared microphones. RootIO uses cloud-based telephony to facilitate conference calls between hosts, guests, and community participants, calls which are broadcast live on air. Based on research that shows an increased participation in cases where the cost of calling is born by an entity rather than the participant [134], all the telephony is designed so that hosts, guests, and community participants receive rather than make calls, which results in them not incurring costs to participate.

I also harness telephony to facilitate self-service messaging on these stations. The most common messages on typical FM radio stations are announcements, usually for funerals and lost items. Greetings are other most frequent type of message, usually by means of a phone call during an entertainment show. For both cases I use an IVR menu to allow users to call in and record messages in three broad categories: advertisements, announcements, and greetings, which are broadcast at specific times on the radio. This in effect alleviates the need to physically go to the station, but also means anyone with access to a phone can leave a message on the station.

Lastly, in communities where I deployed stations in Northern Uganda, voice call minutes were much cheaper than the other services offered by telecommunication companies such as mobile Internet. A mere €0.25 was sufficient to buy unlimited voice minutes for use by the station to call out. Moreover, it is worth noting that while these communities always had a good voice telephony network, mobile Internet was always scant and unreliable. The use of voice calls as streaming links for content from the server therefore opened up the possibility of cloud-based programs in areas where Internet connectivity is not reliable.

Telephony therefore helps in three aspects: 1) facilitating processes that would have required a studio and thereby reducing setup and running costs; 2) increasing access to the station for listeners of the community thereby increasing avenues for input by the community; and 3) facilitating the streaming of content in areas where Internet access is unreliable. Voice telephony emerged as one of the technological tenets of the RootIO setup.

### 3.7.3 Sectorization of the FM space

The most prominent hurdle in setting up RootIO stations is the obtention of broadcast licenses. In Uganda the regulators first claimed a lack of spectrum, however, when the scale of the stations was clarified, licenses were obtained, but the paperwork involved (see Annex B: Licensing requirements in contexts where RootIO was deployed) was much more intensive than is applicable for a typical RootIO station. Even then, the licenses were granted as a regional frequency shared by multiple stations. This had two desirable results.

Firstly, the holder of a license could, with proper planning, install many more micro stations in the region, provided that they do not interfere with each other, which is what ended up happening in Northern Uganda. This results in a sectorization of new stations, which means that FM spectrum is used much more optimally. To this end, I implemented SPLAT4J, an add-on to RootIO that allows operators of RootIO stations to visualize the coverage of their stations given the antenna location and specific broadcast parameters. This allows communities to plan for the best location of the station, but also to evaluate different installation parameters and the resulting impact on the coverage of their station.

The second result is that licensing can be delegated to the holder of a license. Upon obtention of a license, an entity can sub-grant the license, providing that the operators of stations are using the same equipment for which the station is obtained. This reduces the workload of the regulator, but also means that communities without the technological knowledge and lobbying power to interact with regulators have an intermediary that addresses the interests of both parties. In my deployments in both Uganda and Romania, licenses were held by an entity not based in the community, but with the technical knowledge to interact with regulators as well as advise communities on how to stay within regulations.

### 3.7.4 Addressing Skill and labor shortfalls

A typical FM Radio station will need editorial staff and staff for operational activities in business and technical areas, as well as ensuring regulatory compliance. While this is tenable in a metropolitan setting, in rural settings there is a paucity of these kinds of personnel. During my research in Northern Uganda, even when these people were available, they were in high demand by numerous NGO projects and could not devote as much time as would be necessary to run a typical FM radio station. Such was the scale of this shortage that two of the volunteers who launched their careers with the station in Patongo would go on to be hired by regional radio stations as presenters.

The web portal allows for the centralization of activities pertaining to the smooth operation of FM radio stations. A single user can take on a single category of operations for a network of stations, such as sourcing content, while another user focuses on monitoring the technical performance across all stations. RootIO thereby provides a tool that allows for the management of multiple stations using fewer resources. Other innovations such as self-service message menus facilitated by telephony, and content syndication that I will discuss shortly, also help to lighten the labor load on these stations.

The technical labor pertinent to FM radio components is much harder to find, even in city centers, and was non-existent in Northern Uganda. When transmitters did eventually fail due to loose cabling, the attempted repairs evidenced a complete lack of knowledge in the field of radio frequency electronics. The RootIO proposition includes in its pricing model an insurance plan for technical failures. This allows communities to pool resources for technical fixes and equipment replacement, a feat that is hard to achieve for a single RootIO station in isolation.

### 3.7.5 Content syndication and auto-production (TTS).

While the overarching goal of the RootIO design is to increase input from grassroots communities, there is no desire to isolate these communities from information originating from outside. Core to the design of the RootIO platform is the dependence on other media houses and sources for as much of the content as cannot be produced within the community. This, as noted before, lessens the labor load on the station. Options such as subscribing to podcasts and streams means that by performing a one-time operation, a station can have a constant syndication of content from a source that is frequently populated.

Another game-changing feature is the integration of TTS technology into the content generation process. TTS is available as an option for generating the voice prompts that are played when listeners call into the station, either in a talk show setting, or to leave a voice message by navigating an audio menu. In the content section, TTS can also be used to read out, for example a news story copied verbatim from a news website, or any other written information that can be entered as text. Research running along the RootIO project in Romania, Ireland and Madeira yielded a model for incorporating weather information into the content pool by way of TTS [96].

The drawback is that the support for TTS, while very versatile for international languages such as English, is less satisfactory for relatively less spoken languages like Romanian, and virtually non-existent for most of the world's languages. Independent research in projects that RootIO was part of made progress towards TTS generation techniques for under-resourced languages [93], but in my case, all the communities in which I deployed in Northern Uganda were already consuming a fair amount of content in English and were demanding content produced by the BBC and other prominent international media houses, mostly produced in English.

# Chapter 4

## Contexts of deployment: Northern Uganda

This chapter presents findings from running FM radio stations in Northern Uganda in a bid to formulate the specification for a low-power transmission technological stack. I present the benefits realized, challenges observed, and lessons incorporated in the design of the subsequent RootIO stack.

The sites that I studied in the first setup were situated in Northern Uganda. This was because in initial discussions the regulator identified northern Uganda as the one region where FM frequencies were available. The sites selected for station installation were hosting computing centers facilitated by UNICEF. In my initial assessment with UX designer Elizabeth Monacello Gin, we favored the sites that demonstrated strong community presence, judged by activity at the centers. Out of 8 centers surveyed, 4 centers were chosen given that the communications regulator in Uganda permitted transmission at only 4 specified points.

The initial RootIO deployment sought to prove a core concept: that a low-power computing device, coupled with a low-power FM transmitter, could deliver audio quality undistinguishable from that of typical FM radio stations. As such, the deployment in Northern Uganda was geared towards establishing a minimum viable hardware stack for FM transmission and understanding the various opportunities and challenges around which to design the RootIO platform. The lessons learned from this phase shaped the RootIO software platform and helped to identify constraints in the hardware design and the corresponding implications in the software domain.

### 4.1.1 Methodology

Of the four stations deployed in Northern Uganda, I followed two closely and they form the basis for my study. The information upon which my thesis is built was obtained in structured interviews, focus group discussions, monthly reports and a survey within these two communities, and I explicitly specify when information was obtained from, and observations made in, the other two communities. One of the stations, based in Amuru district, was situated at the site of the local government office hosting other initiatives such as a computing center from UNICEF and a community Internet center from SINFA. The other station was situated at the location of a CBO in Agago district, which also hosts a computer center donated by UNICEF. Below I detail the two installations and observations during a four-year period

Given that in the very first concept there was very little interaction between the transmission sites and the cloud-based station management system, all data about deployments was obtained through regular communication with representatives from the communities. Quantitative Technical details about the performance of stations were obtained by voice phone calls and text messages to technical contacts for each of the stations and compiled into a monthly technical report.

Similarly, for qualitative data I relied on monthly station reports focused on programming and personnel aspects compiled by a team from NUMEC, a partner institution that was relied on to provide journalistic and media oversite to the stations set up within these communities.  assistant to communicate regularly with community representatives to log program performance and technical issues.

Further qualitative and quantitative information about the operation, relevance and utility of the station was obtained through a focus group discussion at the station in Patongo facilitated by NUMEC.

This session brought together 5 members of the community that were working at the radio station as well as 2 members of Gwokke-Keni, the CBO that was hosting the station.

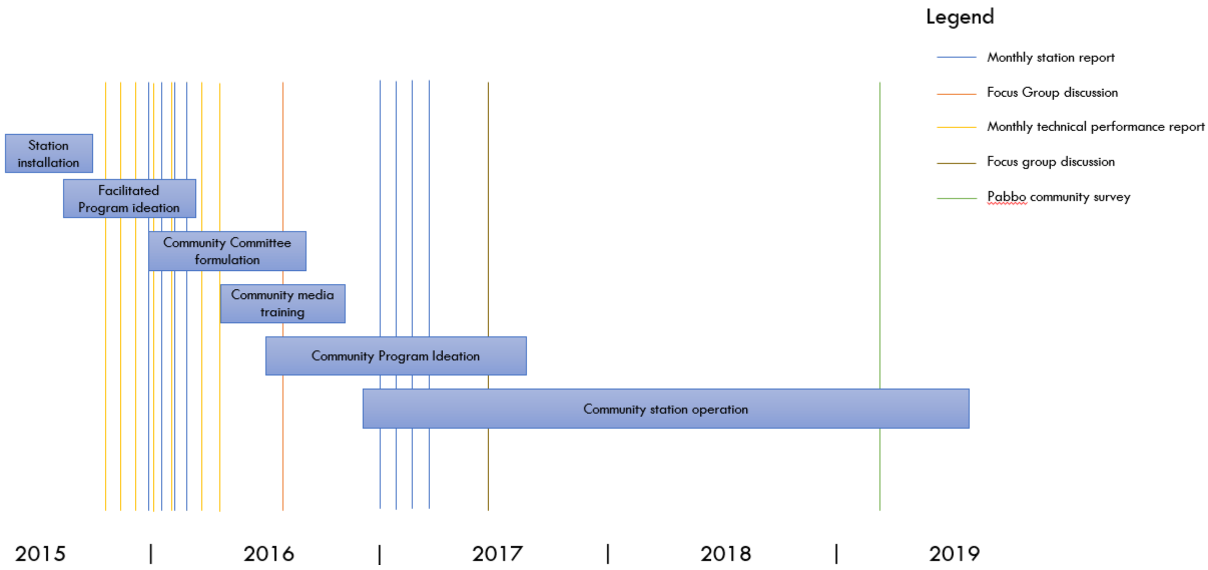



**Figure 4.1: Graphic showing research activities in Northern Uganda**

Towards the end of the period during which I studied the stations, a survey was carried in Pabbo, one of the communities to understand the energy and equipment access as well as media access and consumption patterns within this community. It was assumed that the station covered an effective area of 254.5 square kilometres (radius of 9KM) of the total 762.2 square kilometres of Pabbo sub-county. The station was also assumed to cover 18,179 residents given the total population of 54,415 people reported by the sub-county office. The targeted population was the adult population above 14 years, which is 54% of the Ugandan population, and thereby prorated to 9,448 residents. The survey was carried out among 105 participants with a 1.89% margin of error and a 95% confidence level.

## 4.1.2 Pabbo sub-county

Located in the Kilak North county of Amuru district in Northern Uganda, Pabbo town council morphed out of the largest internally displaced persons (IDP) camp during the LRA insurgency [32] that ravaged the northern and eastern parts of Uganda from 1987 to 2006. This sub-county has a population of 54,415 according to information supplied by the sub-county office. While many families returned to their original settlements after the conflict erupted, most families remained and constitute what is now Pabbo trading center.

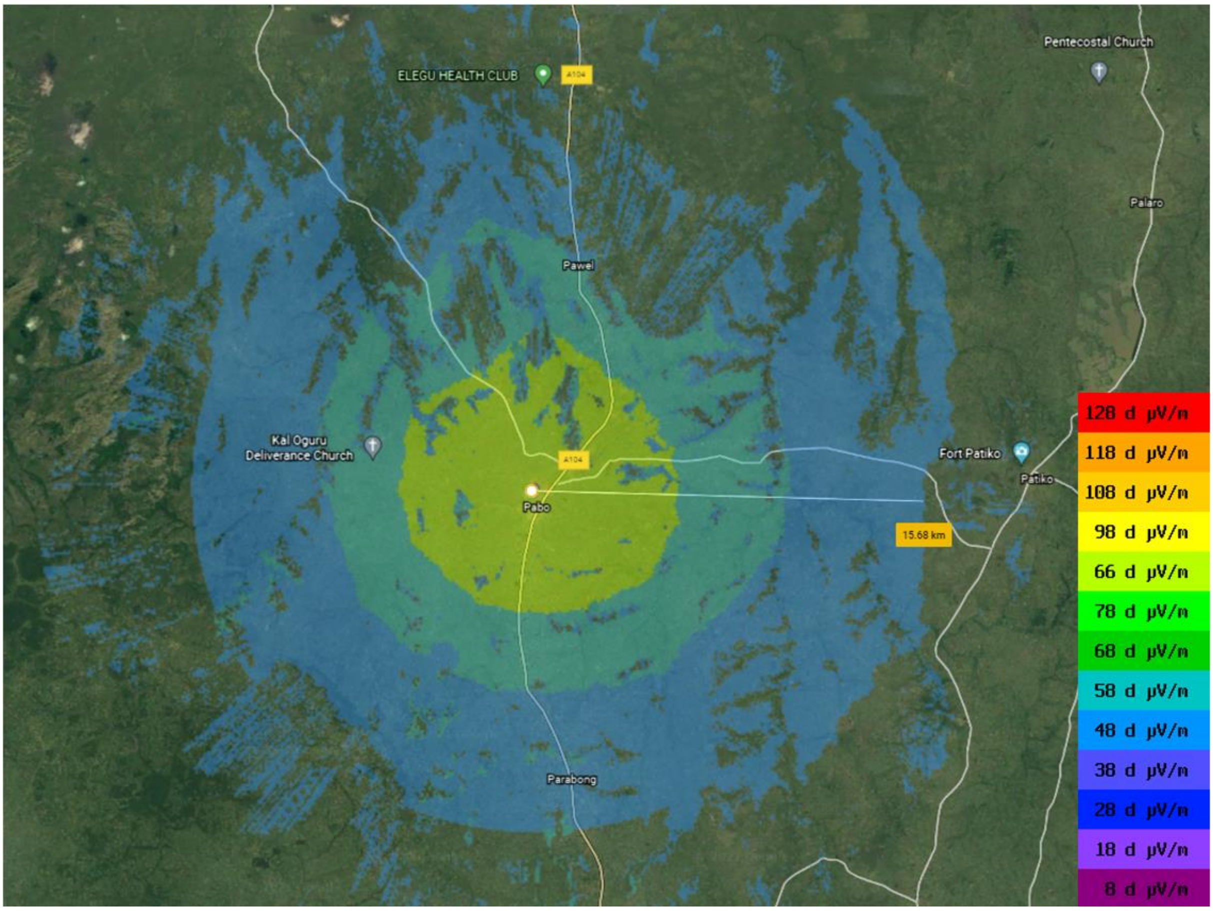



**Figure 4.2: Map snippet showing contours of signal coverage for the station at Pabbo**

In a survey that I conducted among 105 respondents, the main livelihood activities reported were farming of seasonal grains and cereals, and small-scale businesses. Informal employment and farming accounted for more than 60% of daily activity. As can be seen from Figure 4.3, ignoring students who registered no income, farmers posted the lowest daily income with business owners and informally employed individuals registering higher incomes.

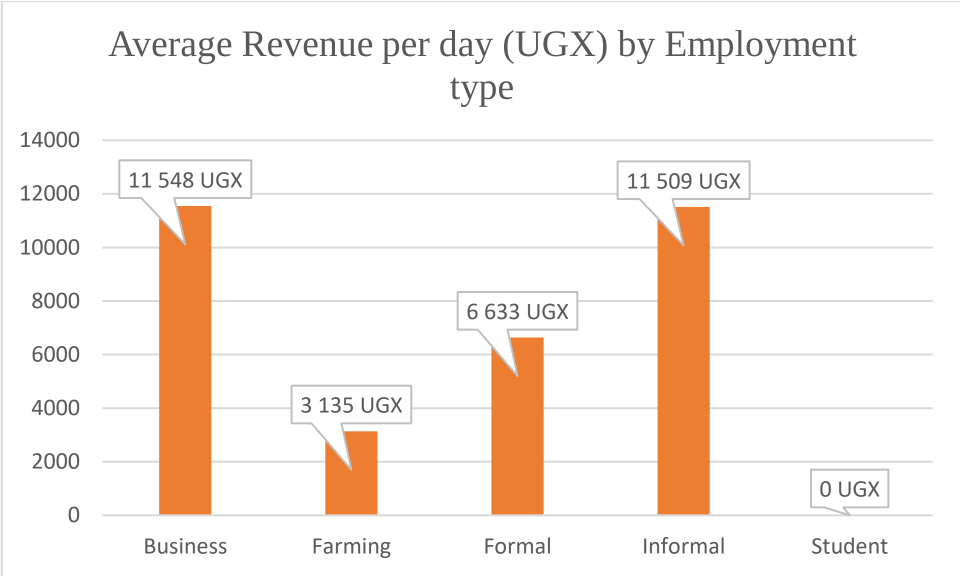



**Figure 4.3: Graphic showing average income across employment categories**

Given the rural situation of Pabbo town, there was scant use of grid electricity at a household level. Most of the electronic devices used in homes (lighting, phones, radios) were powered from dry cells where possible, or from rechargeable wet-cell batteries either charged by solar panels or from a location with access to grid electricity. As such, energy-intense activities, such as cooking, were undertaken using firewood and charcoal that are in abundant supply. While a higher number of correspondents

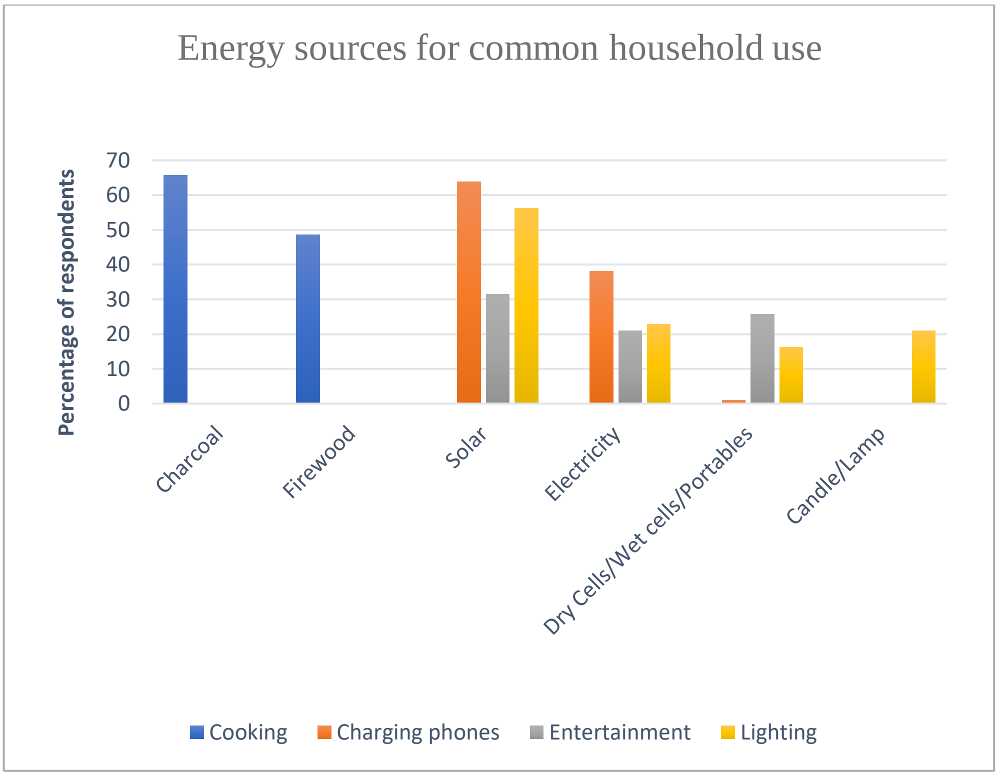



**Figure 4.4: Energy sources used for various household needs**

reported the use of grid electricity for charging phones, it was observed that this is usually done at central locations such as shops and telecommunication agents' booths, where a small fee is paid for the charging service.

Regarding information consumption, the survey revealed that Radio is still, by some margin, the most used medium, and was accessed frequently using mobile phones. 70% of respondents were using FM radio for all information access compared to less than 25% who used television. Though television had a sizeable use, in my travels in these communities I observed that most people watch it from communal locations such as bars and shops in the town center as opposed to within households as is the case with FM radio. This is corroborated by low television ownership rates as can be seen in Figure 4.5.

FM radio was the most owned and accessed communication technology, followed by feature phones, while ownership and access to televisions and computers were still low. Fixed network services, such as landline telephony and fixed internet, were virtually non-existent.

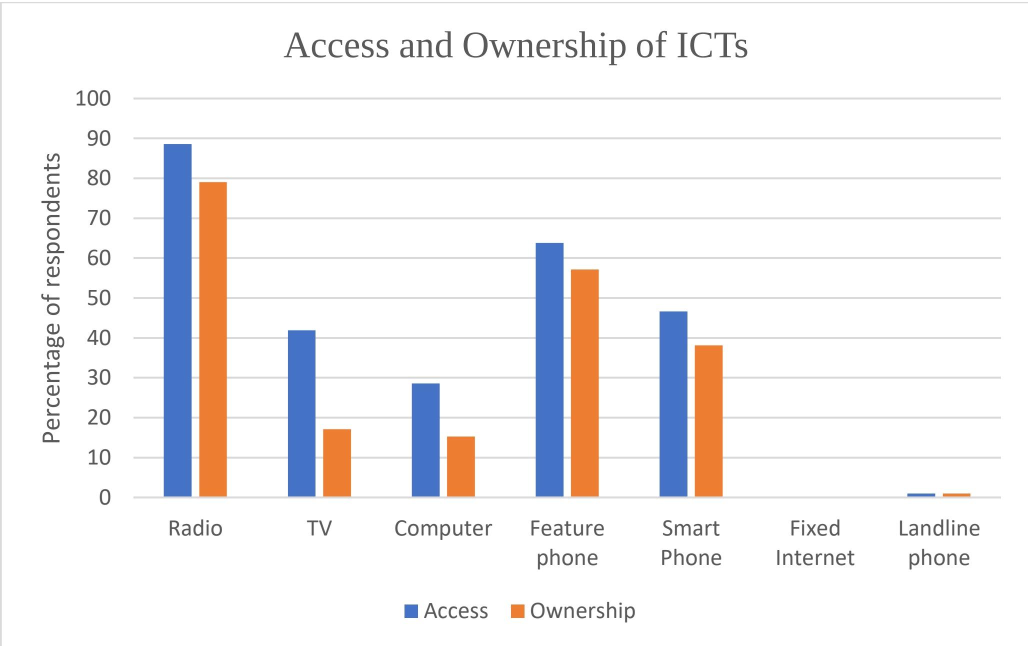



**Figure 4.5: Graphic depicting ownership and access of ICTs in Pabbo**

Regarding mobile phones, the use of native voice and SMS functions were predominant, but the use of mobile Internet is growing. This, however, is due to the fact that voice telephony and texting features have also been moved into the Internet space by OTT applications such as WhatsApp and Facebook Messenger. As such, most of the users of mobile Internet listed Facebook and WhatsApp as the predominant services they access via the Internet.

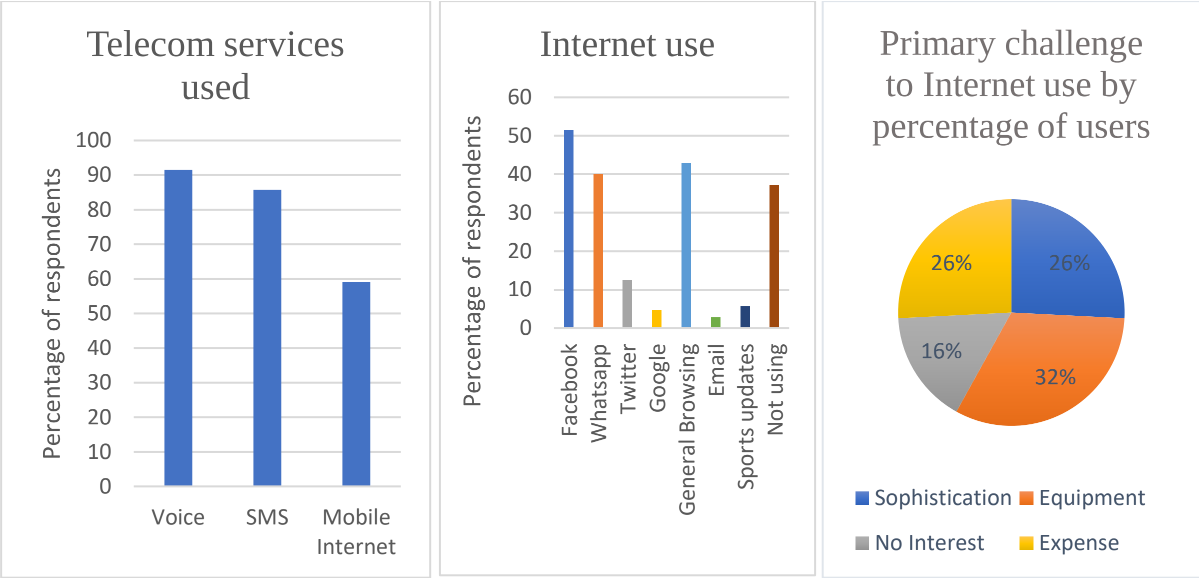



**Figure 4.6:Usage of telecom services (left) with a focus on Internet use (center) and primary challenges to Internet Use (right)**

37% of respondents reported no use of the Internet and the primary challenges listed included a lack of equipment, sophistication, and the expenses incurred in accessing the Internet, conforming with previously identified encumbrances to Internet use in sub-Saharan Africa. In my survey, respondents reported daily expense on telecommunications as a significant percentage of their daily earnings, with the only logical explanation being that from time to time they have to choose between telecommunication services and other basic needs, given the relatively low earnings in these communities.

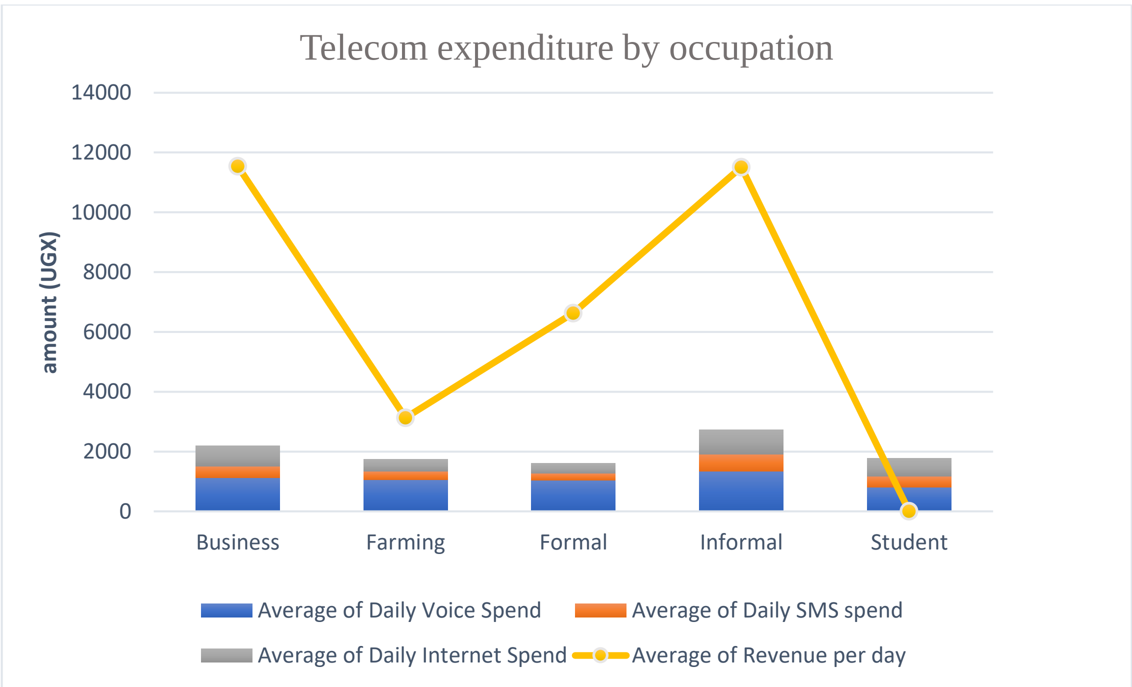



**Figure 4.7: Graphic showing daily spend on telecommunication services vs. daily income**

Individuals in the business sector reported spending up to 19% of their daily earnings on phone expenses, whereas farmers reported spending up to 58% of their earnings on phone expenses. Indeed, Carmody [18] argued that the introduction of telecommunications designed for other environments to developing communities has created new forms of poverty. This was brought out by my research. It is worth taking into consideration that in many of these communities, farmers depend on free firewood for cooking fuel, free water from neighboring streams, stay in houses on their own land, and grow their own food. As such, their ability to meet basic needs outside the monetary economy means they can afford to spend a considerable sum of their earnings on phone expenses.

### 4.1.3 Patongo town

Patongo town forms part of Agago district and was one of the areas most affected by the LRA insurgency that ravaged Northern Uganda for two decades. Given the aforementioned constraints that mandate the colocation of RootIO installations within organizations in the community, I was looking for a centrally located entity that was familiar with ICTs as well as carrying out activities focused on the community.

The ideal partner emerged as a local CBO that was attending to the needs of women living with HIV/AIDs. The CBO, aptly named “Gwokke-Keni”, meaning “look after yourself”, hosted a youth center facilitated by UNICEF. This was an ideal location for the colocation of the station given that it already accommodated solar-powered ICTs and was part of a network of youth centers receiving IT support from BOSCO Uganda.

At the time of installation, the whole of Agago district had no radio stations and was relying on regional radio stations based 100 kilometers away in Gulu, the regional capital. To broadcast for a single hour to their listener group, the team from the host CBO had to travel to this regional capital for an hour-long slot on the radio. This process presented two major encumbrances: firstly, the program was targeted at a very small section of listeners of the station that they were using and secondly, an hour on the station cost in the region of 200€, bringing the cost of a single broadcast to roughly 600€ after paying travel and accommodation costs.

The setup of a RootIO station with this organization had therefore the double benefit of saving the organization money when there was need to communicate, as well as limiting the broadcast to listeners to whom it was most relevant. As we will see later, the community turned the station into a business with signs of profitability before having to cease operations due to the expiry of the trial license on which they were operating.

### 4.1.4 RootIO station installation

The RootIO stations installed at Pabbo and Patongo were two of the very first stations installed by

RootIO and therefore featured mostly antiquated hardware (CZE T251A transmitters, fixed steel towers, bucket enclosures and older smartphones). These stations, initiatlly prototyped by Csikszentmihalyi to be housed in ubiquitous polypropylene buckets, were assembled in partnership with the Log'el project, a local electronics training initiative for people living with disabilities. The aim was to have a local entity capable of assembling these stations in the event of future local demand.


**Figure 4.8: RootIO station buckets being assembled by the Log'el project for the Ugandan deployment**

The tower was fabricated in the capital, Kampala, and cost 1200€ per station. Given the weight and specifications of the tower in this particular installation, the design, manufacture, and installation of the tower was itself a sub-project handled by a contracted entity. This was identified as an obstacle that needed to be eliminated in subsequent installations. Similarly, the installation of solar equipment including panels, batteries, charge controllers, and charging outlets was handled by a separate sub-contractor, but this was not regarded as a challenge owing to the prevalence of solar energy installations in rural areas of Uganda, and therefore solar technicians were also prevalent.

My primary responsibility was the installation of the core RootIO equipment made up of the equipment in the bucket, the RF transmission cable, and the antenna. Given the studio-less nature of RootIO, all that was needed at the transmission sites was a room to host the solar batteries with proximity to the point where the bucket was being mounted. Conforming to the desired goal of easy deployment, this installation involved minimal work, namely: cable termination and trunking, mounting the antenna, and mounting the bucket on the wall. In subsequent deployments, considerations were made to eliminate potential stumbling blocks such as on-site RF cable termination and complex antenna installations that required sub-contracting.

**Figure 4.9: The RootIO bucket for the station installed in Patongo (left) and a panoramic view showing the station building and tower (right)**

Unlike the station at Pabbo that featured a purchased prefabricated antenna, the station at Patongo used an antenna that we installed in a field using cheap aluminum pipes and plastic agricultural tubing. The characteristics of the halfwave dipole antenna were agreeable, demonstrating an SWR of 1.5 which is acceptable for FM broadcast [49]. The initial design in this phase focused on keeping everything local and trying to use resources and materials that were common to the populations in these areas. It was subsequently realized that most of the core components for RootIO stations, such as transmitters and RF cable, had to be imported and therefore manufacturing antennas locally afforded minimal gains while increasing the risk of malfunction, given the delicate considerations involved. The constraint to rely on local materials was relaxed and efforts were instead focused on sourcing materials internationally, such as the telescopic towers and terminated RF cable, that make deployments faster, easier and more error-proof.

## 4.2 Recreating the FM Radio experience in first deployments

Given the prevailing use of FM radio, I tried to recreate this experience using the cheap ensemble proposed under RootIO. It was clear from surveying the crowd that a piece of media playing through the RootIO setup was indistinguishable from media being produced by the big FM radio stations. The challenge was how to string together programs with varying media formats in a way that would provide an acceptable FM radio experience for the communities. As I show later, interaction with communities running these stations resulted in models for various program formats, media formats and sources.

### 4.2.1 News

News shows are a big program item in the communities in Northern Uganda. Recalling that FM radio is in these communities the primary, and in some cases only, channel for mass information access, news programs gain increased importance. In many stations in Uganda news stories are sourced as text files from the Uganda Radio Network (URN) [135], an umbrella association of journalists covering the entire country. The initial options were either to pass the text news file to someone in the community who would then call into the phone at the station and read the news file or have someone record a reading of the news file and upload the resulting audio file to the cloud server and have it played on the station phone through a phone call from the server.

The former option presented an opportunity to get local talent involved in station activities but was severely impeded by scant Internet connectivity (the file is downloaded from the URN website) and the lack of available devices to facilitate the file in a readable format. Also, given that the station was relying on volunteers, it was hard to guarantee commitment to this task. Paying permanent staff for this task would be more expensive than procuring the service from NUMEC who were already in the business of producing media for other radio stations. In the end, the decision was made to have NUMEC record a reading of the news and drop the audio file in a Google Drive location, from where it was then picked by a script on the RootIO server and played at the time when a news program was scheduled on a particular station.

Given that news is listened to predominantly for informational purposes, using telephony did not severely hamper the purpose of news broadcasts, especially taking into consideration that the most common G711 and G729 telephony codecs are designed to best convey the human voice [54,55]. The files were being recorded in a studio setting, which meant there was no ambient noise as is typical with phone calls, and the resulting audio was significantly better than someone speaking into a phone. Eventually, this design suited the use scenario for news programs.

### 4.2.2 Talk-shows

Talk shows are another major program format on local FM radio stations. These are usually topical and feature a resident host who discusses a subject of interest to the community, optionally with an expert. Typically, these will offer the possibility for members of the communities to call in and express their opinion, challenge views expressed, or ask questions. In these scenarios the host and invited expert will usually be in a studio, something that RootIO stations did not possess.

The initial approach in Northern Uganda would see the host and invited guest sit next to a phone that was called by the cloud server when it was time for their show, according to the program schedule. The host and expert would discuss freely, and their conversation would be aired on the radio. This model offered no easy way for members of the community to participate in talk shows and this was also identified as an issue that needed addressing in subsequent designs.

Given the spoken nature of talk shows, it was also clear that they were listened to predominantly for informational purposes, diminishing the significance of audio aesthetics. I therefore implemented the first talk show as a conference call between the host, the cloud server, and the transmission site. Given the lack of a proper web portal at the time, I also made accommodations for hosts to call directly into the phone at the transmission site to get on air. In this scenario, intended participants would call the host who would then add them to the ongoing call in a conference setting. The audio samples that I listened to, however, bore significant levels of ambient noise, and in some cases the audio was faint because the host was too far from the phone, or significant breathing sounds became audible as the host tried to speak straight into the phone for louder audio.

### 4.2.3 Announcements

In regions in which Internet-based social networks have less penetration, FM radio announcements are still a primary mode for disseminating important notices. In the communities in which I worked in Northern Uganda, announcements were among the most requested features of the radio station. In these communities, these announcements are predominantly funeral announcements and, in some cases, informational notices for events happening in the communities, such as immunizations, training opportunities, and other events that involve mass community participation. These announcements are usually aired in the early hours of the evening.

In a typical FM radio station, the announcements are read by a presenter in the studio and therefore also fall into the category of spoken content and therefore the essence of audio aesthetics is diminished. In the Northern Uganda deployments, I observed the practice of people walking to the premises of the station with a written announcement, or they would narrate their announcement to one of the volunteers at the station premises and pay a small fee for the announcement to air. The volunteer presenter would later call into the station and read the collected announcements to the community.

This practice proved functional for the community, and a review of the income for the station in Patongo revealed that announcements generated 4% of the revenue of the station. I identified this as a practice that necessitated the need for a physical studio, an artefact that the RootIO model sought to do away with. This informed the self-service IVR menu that allowed community members to leave their messages without having to walk to the station.

### 4.2.4 Music

Of all the content that was being aired in the first deployments in Northern Uganda, music programs were the hardest to implement. On a typical FM radio station, music programs take up the majority of airtime. During the night, when studios are not manned, the airtime will consist entirely of music, punctuated by advertisements and station jingles, whereas day-time schedules also feature music programs, though with sporadic speech sessions by presenters.

On RootIO stations the music had to be categorized into different playlists that would then be scheduled in different programs at different times of the day. This was done by maintaining an SQLite database file with mappings for songs to playlists, playlists to programs, and programs to specific time slots on selected days. The RootIO app would then consult the SQLite database at different times to determine what content to play. This meant that adding new songs or modifying programs could not be done by people within the community given the lack of computers and even greater lack of knowledge regarding how to deal with SQL databases.

Playing Music through phone calls was ruled out first due to the quality degradation that was bound to occur, as well as the cost it would have incurred. I would therefore assist communities with this mapping and pass back a file that was then installed on the phone. It was imperative that an acceptable method for managing playlists, programs and schedules be instituted. This too was identified as a design consideration if RootIO stations were indeed to offer a solution that could thrive in an environment with diminished computer skills. This informed the design of playlists, which is detailed in sections 3.4.3 and 3.5.5.

### 4.2.5 Jingles

Jingles are short audio messages that air frequently on radio stations announcing the station name, the frequency, and the tag line for a particular station. Jingles are foundational because they differentiate stations, but are also catchy and often emphasize the station’s genre and identity. There was no easy way to implement jingles in the initial setup given the timing requirements. The initial design involved including a couple of jingle files in playlists and leaving their play to chance, given that the songs on the playlist would always be played in a random order.

I also encouraged the volunteer presenters to mention the station name and station frequency as often as they could, whenever they were presenting something on the station. Subsequent implementation, as detailed in chapter 3.4.4, resulted in an approach in which predefined jingles are played at configurable intervals with a quality similar to that of typical FM radio stations.

### 4.2.6 Advertisements

Advertisements are the main source of revenue for FM radio stations. In typical FM radio stations, pre-recorded adverts punctuate programming at pre-defined times. Given that these adverts are in most cases pre-recorded and have musical effects to sound catchier, these too presented a challenge because the quality would be lost if they were played through a phone call. In the early stages the volunteers at the stations would, in some cases, swap the audio pin from the sound output of the phone into a device playing the adverts, and would swap it back into the phone once the advert concluded. Another approach was for the volunteers to read out the advertisements whenever they would be speaking on the station and in the process of reading news, announcements, or interacting with someone in a talk show setting.

In the station in Patongo, advertisements accounted for 16% of all earned revenue, as seen in chapter 4.4.5. Given the importance of advertisements, it was important to provide a mechanism that allowed advertisements to play with a certain audio quality and in a way that allowed stations to account for the times when advertisement had been played. Subsequent designs catered for the timing and accounting issues and in some cases, where streaming is employed, stations can play advertisements in their full quality as opposed to playing them through telephony.

# 4.3 Observed challenges in the early deployments

As mentioned earlier, the very first deployments in Uganda were geared at assessing the end-to-end viability of a cheap broadcast infrastructure and therefore featured very little of the cloud infrastructure and functionality found in the present RootIO setup. As such, most of the observed challenges in the first deployments were mostly concerning hardware and sound quality, as well as suggestions for other necessary features. Below I recount the major issues that were reported following the launch of the station and what was done to address them.

## 4.3.1 Harmonics in audio transmission

Professional FM radio stations will often feature an expensive audio filter for harmonics, but this was not tenable given my cost and power constraints. The initial stations featured audible harmonics associated with electronic circuits in the bucket housing the station components, as well as the wireless communication by the mobile phone in the bucket. Following recommendations from Vijayaraghavan et al [112], running the audio through a cheap ground loop isolator was enough to remove the harmonics induced by electronics, while shielded cable provided protection against all the noise induced on the audio line.

It was also observed that most listeners were receiving the transmission on cheap FM receivers and mobile phones, in line with the prevailing economic and energy deficit. The use of these devices with relatively limited audio frequency range resulted in communities being more tolerant to slight levels of electronic noise because it was barely perceptible on the devices that they were using.

## 4.3.2 Transmission line failures:

The cable employed in these settings was heavy duty and durable, but frequent failures were encountered with the connectors. This was partly attributed to tampering, but also in some cases the cabling was loose, resulting in movements that would eventually loosen the connectors and cause the failure of the transmission line and prompt the transmitter to shut down. All but one of the reported hardware failures in the Northern Uganda deployment were due to failure of the transmission line. The lack of technical knowledge in these communities to repair the connections mandated that subsequent installations use prefabricated cable with crimped connectors, as opposed to reusable screw-on connectors. It is also preferred practice to have a spare cable that can be swapped in, rather than try to repair the damaged cable.

### 4.3.3 Coverage mismatch

Compared to typical FM radio stations, RootIO stations have very small geographic coverage. While typical stations emit power in the range of kilowatts, RootIO stations emit a paltry 25W at most, and it is not far before their power drops below the recommended minimum for useful FM reception [81]. A common complaint in these areas was that the station was not covering some neighborhoods considered part of the community for which the station is installed. In some cases this complaint was generated as edge communities that wished to be included, a problem of expectation that is not technical in nature.

Considering the nature of RootIO stations, it is imperative that they build on existing infrastructure in populated areas, given the increased cost and risks of deploying them in isolated, but ideal, transmission locations such as hill tops.  SPLAT4J, my open-source adaptation of the popular SPLAT [71] mapping tool, gives a good indication of the areas to be covered by a broadcast given the transmission properties (antenna location and height, ERP, frequency, receiver height). Subsequent deployments in Ireland and Romania simulated the coverage based at multiple transmission points and were able to pick locations that gave optimal coverage of the communities of interest. Finally, the project does not seek to increase coverage are, but rather to accommodate edge communities with their own micro-stations.

### 4.3.4 Physical tampering

Given the limited functions that the initial RootIO deployments came with, and the lack of a studio, communities had to innovate and improvise ways in which to use the station to the benefit of communities. Communities introduced audio inputs that would allow for a studio-like setting and added hardware that allowed them to achieve the necessary setting. This resulted in disruption of the system functions and hardware failures in some cases. The devices were also introduced with no regard for the capacity of the solar batteries, resulting in their degradation due to overly draining them.

It became clear that the communities had expectations of “proper radio,” were being let down by the feature set offered by the nascent system, and were trying to improvise to create a radio experience akin to what they were used to. Subsequent iterations of the RootIO platform feature more input options to allow for the community to contribute content to the station, as well as options to automate most routine station operations.

### 4.3.5 Political issues

Given the centrality of FM radio in the sub-Saharan Africa media landscape, it was inevitable that control over the station would not be excluded from local politics. We did receive reports where some community members were ostracized from management of the station on political grounds. In pursuit of balance, as is proselytized by UNESCO [39], I not only proposed that RootIO stations are governed by a multi-stakeholder committee, but also subsequently included an interface that allows for the tracking of decision-making to gauge the degree to which the community is involved in shaping the nature of broadcast on these stations.

The station in Aber, one of the other locations in Northern Uganda, was shut down by the local leadership on the grounds that the operators of the station were not remitting revenues earned to the local government office, a requirement without basis in broadcast regulation nor the MOU signed with the local government office that was hosting the station. In the station in Pabbo, attempts at reopening the station after obtaining licenses were futile given that the most passionate volunteers were not politically aligned with the sub-county office that was hosting the station, and yet the staff in charge of the station at the local government office lacked the skill and commitment to operationalize the station. I also observed that SINFA, BOSCO and UNICEF had withdrawn their ICT setups from the local government premises.

The station in Patongo, on the other hand, fared better because it was situated in an organization dependent on community appraisal to survive. The fact that this organization also needed to raise its own funding must have driven its initiatives to make a viable business out of the station. The lesson learned here was to not co-locate stations with political government offices, but preferably with entities that survived on community appraisal, had a commercial need, and ensured that the dissemination of information was a large part of their agenda. For these latter entities, the FM station was an asset as opposed to being a would-be artefact for the former entities.

### 4.3.6 Stringent licensing requirements

FM radio is still heavily regulated in many parts of the world, no less in sub-Saharan Africa, and regulators prefer to err on the side of caution rather than entertain experimental approaches as proposed by RootIO. Obtaining FM licenses for RootIO stations in Uganda took slightly over a year from the date of first request and involved lobbying from trustworthy institutions. Unfortunately, many communities do not possess the engineering knowledge, and lobbying capacity, to confront regulators.

Even when licenses were obtained, in many cases those granted are much larger than would be necessary for stations at the scale proposed by RootIO. The waiver of requirements is non-committal, exposing stations to regulatory audits in the future. The same is true with price waivers; stations are at the mercy of regulators and any demands for the full license price would result in stations shutting down.

This is what inspired the SPLAT4J add-in, potentially allowing for license sharing where modelling demonstrates a lack of interference using multiple transmitters within the area which the license covers. This would at least enable multiple stations to afford the full cost of the license.

As explained in section 1.5.1, communities using RootIO have made a case to regulators in seven countries, successfully obtaining broadcast licenses without fulfilling all the laid-out requirements. This is in line with one of my founding theses that the reduction in size of broadcast domains can lead to a relaxation of long-held constraints in FM broadcast, opening up new possibilities for FM radio at the grassroots level.

### 4.3.7 Equipment paucity

Even though RootIO strips radio stations to just the FM transmitter, the bare components necessary to setup a RootIO station are barely present within rural communities that would best benefit from these setups. Some equipment, such as FM transmitters, RF cables, connectors, and antennas could not be sourced even in the capital city of Uganda, inhibiting community-led installations of these stations. Even equipment that is in relatively abundant supply such as solar panels and batteries, are in many cases counterfeit and require a significant amount of diligence to procure and deliver.

Solving the supply chain problem is far beyond the scope of my thesis, but my novel contribution is a design of a low-power computer with an integrated FM transmitter, allowing for an off-the-shelf product that alleviates the need for sourcing multiple components. Having ported the transmission site software to a version capable of running on a Raspberry Pi, the next step would be to integrate the Raspberry Pi with an RF board, such as those supplied by PCS electronics, one of the FM transmitter manufacturers I used. Even then, these basic components still need to be imported. I also assessed the design of antennas from readily available agricultural tubing and aluminum tubes, the design of which allows for installing antennas within local communities, albeit requiring individuals with some technological skills. For the foreseeable future, distanced communities will have to rely on an intermediary to help with the procurement of FM broadcast equipment.

## 4.4 Benefits of the RootIO station to local communities

Operating the station in the community of Patongo yielded several benefits to the local community and revealed novel possibilities, given increased access to FM broadcast coupled with an appropriately sized listenership. Besides generating revenue for the CBO that was hosting the station, gains were also registered in social circles. Below, I share highlights drawn from focus group discussions and email exchanges with volunteers who were manning the station.

### 4.4.1 Redefining the center

By focusing on women living with HIV, Gwokke-Keni was addressing a community far-removed from the center stage and that was subject to stigma. This is another example of a community that is socially distanced, and that risks being distanced in other aspects as well. The members of this organization highlighted, in one of the focus group discussions, that the station transformed the organization into an important entity in the community and became a meeting point to discuss various topics in the community.

People who came to leave announcements and adverts on the radio, as well as the youth who were planning and running various programs on the radio station, as well as and the youth center, gave the organization a vibrant atmosphere. This was in stark contrast to the previous perception of the organization as a meeting place for mostly widowed women. This was a clear example where a society that was cast to the peripheries by traditional broadcast media, ended up at the center within the community of its relevance. This resonates with current trends in on-demand music and video streaming platforms where genres and artists, who would not stand a chance on traditional broadcast media, are now able to have their own space and followers.

### 4.4.2 FM radio for unlikely participants:

The diminution of FM stations as implemented by RootIO in Patongo fostered access to FM radio by groups that did not use FM radio because it was expensive, far removed from their contexts and addressed a listenership much larger than their intended audience. In Patongo two cases stood out: the first was that of a farmer who used the FM radio to announce and recover their lost cow -- something that they would not have done on one of the larger stations. The second was a local secondary school whose advertisements on the RootIO station in Patongo resulted in markedly increased admissions from within this community. A survey of participants on the radio revealed that the group with most participation were female traders within the community advertising their merchandise, a stark contrast to typical commercial FM stations.

### 4.4.3 A localized flavor of FM radio

Handing the control of FM radio to these communities allowed them to weave their local norms into the fabric of the station, giving it a local relevance that contrasted with what we saw with the bigger FM radio stations. In one instance, for example, it was decided that funeral announcements would be free of charge as an act of empathy with grieving families, in contrast to typical FM radio stations that charge for these announcements.

Similar unique use scenarios were observed, such as live narration of hyper-local football matches, as well as live broadcast of local government meetings. This was made possible by diminishing the scale of the radio station to the size of local governments and handing control to people within the community. These examples align well with my hypothesis that reducing the broadcast size unlocks the potential for novel use of FM radio by grassroots communities for intra-community communication.

### 4.4.4 A ramp for media practitioners:

Typically, for presenters to be employed on radio stations in Uganda, they need to have studied journalism in a tertiary institution. Working with NUMEC, communities were taken through a "barefoot" journalism course in which participants from the community were taught the basics of broadcast. From this approach, the station in Patongo was able to build a corpus of 12 volunteer presenters that formulated a schedule with 15 distinct programs per week. Some of the volunteers that worked with the RootIO station in Patongo were eventually hired by bigger FM radio stations in the region, in line with my thesis that RootIO stations can bootstrap an electronic information culture in distanced communities that brings them closer to contemporary digital technologies.

### 4.4.5 Revenue opportunities

The station in Patongo generated its revenue mostly by selling advertisement and announcement space on the station, at a price substantially cheaper than large FM radio stations. Another revenue channel was the sale of airtime slots to participants who intended to use the station to communicate within the community, such as campaigning politicians. Additional unconventional revenue sources included membership fees and "on the spot" messaging.

**SUMMARY OF INCOME FROM JULY 2015 TO JULY 2017**

| CODE | Details | |
|---|---|---|
| 111101 | Subvention Grant | 4,500,000 |
| 111102 | Adverts | 1,600,000 |
| 111103 | Announcements | 442,000 |
| 111104 | Talk shows | 2,100,000 |
| 111105 | Sport Massages | 600,000 |
| 111106 | Membership fees | 400,000 |
| 111107 | Annual subscription fees. | 400,000 |
| **Grand Total** | | **10,042,000 UGX** |

**Figure 4.10: Excerpt from a report from the station in Patongo showing earnings over a period of 2 years**

At one point, the station had $140 on its account, and generated slightly over $2,700 in the space of two years. It is worth noting that the setup cost for the station was appropriately $3,200 (in the prevailing exchange rate at the time). The station therefore contributed significantly to the revenues of the hosting organization and paid for some services and labor that were used for the organization's other activities, fomenting a synergy that increased support for the station locally.

## 4.5 Lessons and outcomes of the installations in Northern Uganda

The deployments in Northern Uganda, as previously mentioned, were geared at assessing the possibility of an FM broadcast using minimal equipment and powered by renewable energy. These deployments would provide useful inputs for future hardware designs and program formats. Below I list the outcomes and lessons learnt from this deployment.

## 4.5.1 A new licensing regime

The initial licenses for the deployments in communities in Northern Uganda were trial licenses and were therefore not clearly defined in the existing licensing regime, making it impossible for communities to replicate these deployments in the future. The findings of the RootIO trial in Northern Uganda were the basis for the communication regulators in Uganda to grant a new license for FM stations on the scale proposed by RootIO, and these are far more accessible to rural communities, both in terms of fees (200$ for a license covering 10 stations), as well as in relaxing the requirements that are not applicable to these stations.

Constraints still abound, however, in the Ugandan context, constraints such as the need to have transmission equipment certified by international labs to standards EN301489-1 [37], EN301489-11 [35], EN302018-2 [36], EN60215:1996 [16], and 1995/5/EC [136]. The manufacturers of transmitters in the power range that is used by RootIO mostly target an amateur stratum and typically self-certify, as opposed to manufacturers for professional stations that can afford certification by international labs. In the case of the latter RootIO stations in Uganda, it was agreed that test results furnished by the manufacturer (and not necessarily an internationally accredited lab), coupled with a European Conformity (CE) certificate, would suffice for this purpose.

## 4.5.2 A blueprint for content sources

Given the content types that were being requested, it was possible to identify the different sources of content that would satisfy listener needs in the absence of a studio. The needs within these communities ranged from content sourced locally to content produced overseas by international media houses. In all these cases the challenge included sourcing and delivering the content in a manner that is timely, affordable, effortless, and with minimal degradation of audio quality. In the table below, I show the content requests and explain why a particular approach was chosen with consideration for the aforementioned factors.

| **Content requested** | **Content type** | **Approach chosen** | **Explanation** |
|---|---|---|---|
| Announcements from within the community | Audio speech | Dial in to an IVR leave a voice message | Given that this is spoken audio, the emphasis is on the perceptibility of speech rather than audio quality. Calling is also convenient for both the calling party and the administrators of the |

| | | | |
|---|---|---|---|
| | | | station. |
| Local news | Audio speech | Phone call to news reader | Given the paucity of audio recording and editing equipment, it is better to call the news reader and have them read the news in real time than have them record and edit the news file for subsequent upload into the system |
| National news | Speech audio | Upload audio file | The audio news stories we opted for were being catalogued by Uganda Radio Network, and issued as text files on their paywalled portal. We worked with a local journalist association to record these news stories and upload them into an audio repository. |
| International news | Speech audio | Podcast synchronization | These files are created by media houses and published regularly. I use a service to check for new files and syndicate them to the stations on which they are scheduled. |
| Sports updates | Pre-recorded file, downloaded from a podcast repository | Podcast synchronization | Most requested updates were for European soccer leagues. These too are available as podcasts, and are published regularly. |
| Synchronized real time addresses, e.g., presidential national address | Speech audio | Icecast stream played by device at transmission site or through phone call | These addresses are real time and in some cases are streamed by the bigger media houses. The phone can play the content directly from an online stream or, if lacking Internet, a call can be |

| | | | |
|---|---|---|---|
| | | | made to the station and the stream played in the phone call. |
| Music mixes | Music | Local media player in device on transmission site | These files are not changed frequently and can be pre-loaded on the station device and updated occasionally. The platform allows for the creation of playlists from these files. |
| Talk shows | Speech audio | Conference calls to host and participants, also broadcast at transmission site | Given the lack of a studio, live interactions are facilitated by a conference call with one participant acting as the host, with the ability to bring in other participants. This conference call is piped into the station by means of a telephone call. |
| Proceedings of local government meetings | Speech audio | Upload recorded file into media track or make a call to on-site reporter for live streaming | Depending on whether the need is to broadcast these meetings as they happen or later, the meeting can be broadcast through a phone call or recorded using a device such as a phone, and the audio file can be later uploaded to a particular station. |

**Table 4.1: Table showing desired content types and chosen delivery approach**

It became clear that content could be divided into two broad categories: speech and music. Spoken content can be easily relayed through telephone calls, whether via SIP or PSTN. The emphasis is always on the perceptibility of the human voice, something that is well accommodated in all telephony codecs. Theoretically, it is possible to play Icecast streams into phone calls that are terminated at the transmission site; the advantage here being the ability to play Internet-based content at remote sites, even in the absence of an Internet connection. The lack of means to differentiate between speech and music streams led me to not implement this feature because it could easily be misused to play music content, resulting in a degraded experience, especially when the call is delivered via PSTN telephony.

Music and entertainment content contain multiple instruments that are poorly accommodated in telephony codecs, and therefore should be played, where possible, locally on the phone by a more appropriate media player. Given the fact that this content also takes up the bulk of the schedule, it makes sense to pre-load it on the phone when circumstances allow, such as in the case of music files, to minimize dependence on the Internet that is both scarce and expensive. For streams of entertainment content, it is advisable that these are accessed via the Internet and played by the device at the station, in order to avoid loss in audio quality that may result in a bad listening experience.

### 4.5.3 Review of equipment

The initial setup provided an opportunity to identify possible optimizations in the hardware that was being used in the setup of RootIO stations in distanced communities. The aim was to identify the optimal configurations of hardware for grassroots broadcast. The lessons learned spanned from the point of installation to two years post-installation, allowing us to observe the long-term effects of human and environmental factors.

The first lesson learned was regarding the towers hoisting the antenna. Because these towers were made out of steel, they were heavy, expensive to manufacture, and required skilled labor and several weeks to install and erect on a pre-built permanent concrete foundation. This presented major demands in terms of skill, cost, and time. It was therefore determined to source an alternative that would alleviate these incurrences, with a viable solution emerging in the form of fiberglass telescopic towers. The downside to this is that they have to be imported, but this presents a problem in only one dimension and the resolution to this, for example, is by to use of a middle-man agency, which would result in increased efficiency in deployment.

The decision to use fiberglass towers had an implication on the cable that needed to be used for subsequent deployments, with preference shifting to the more flexible, and lighter, H155 low loss RF cable. The heavier cable previously used required sturdier support structures, resulting in increased trunking work during installation. The increased flexibility, and reduced weight, both reduced installation time and incidents arising from cable damage.

The last lesson learned was related to minimizing installation and assembly of station equipment on-site. I observed that the skills necessary to properly install parts, such as antennas, as well as crimp RF cables, were in short supply given how alien these technologies are in grassroots communities. Also, replacement parts and the equipment necessary for RF setups are not readily available and would have to be sourced from the capital city. In later installations, I ensured that all hardware parts were integral and that all that was remained to do at the transmission site was interconnecting the various hardware components.

### 4.5.4 Equipment isolation

The need to collocate RootIO station equipment within the facilities of organizations in the community exposed them to the risk of tampering. This, however, was also a learning experience because it brought to the surface the desired media and program formats that resulted in these actions at the community level. Unfortunately, in both locations, this resulted in damage to the transmission line and the transmitters shut down. Efforts were made to repair the setup in these communities, but my inspection revealed a lack of knowledge of the workings of RF electronics in these communities, and no success was achieved in bringing the stations back on air.

From this, three design considerations were conceived. The first was to provide input mechanisms that allowed communities to go on air without having to open the station setup and connect extra hardware. This was achieved by streaming to an Icecast server with the phone then receiving and playing the stream, or in cases where the lack of Internet does not allow, connecting a simple mixing board between the phone and the transmitter allowing for optional, additional audio inputs into the transmitter.

The second consideration that arose from this was the need for an insurance scheme in the revenue model to replace equipment in these communities when there is a failure, as well as keeping on site spares for some of the cheap and easy-to-replace parts, such as RF and electrical cables. The third consideration was to house the equipment in containers that are easy to store off-site, and for this, rack cases were more appropriate because they are easier to accommodate in out-of-sight spaces. The overarching long-term result was the conception of an integrated transmitter where all the key components were invariably interconnected with less room for disruption of the desired model.

### 4.5.5 Viable minimal stack for FM broadcast

The ultimate goal of the initial deployments in Northern Uganda was to prove that quality FM broadcast was viable on a minimum hardware stack. Even if the programming fell short on many fronts, audio samples sent in by volunteers from the community were of good quality.

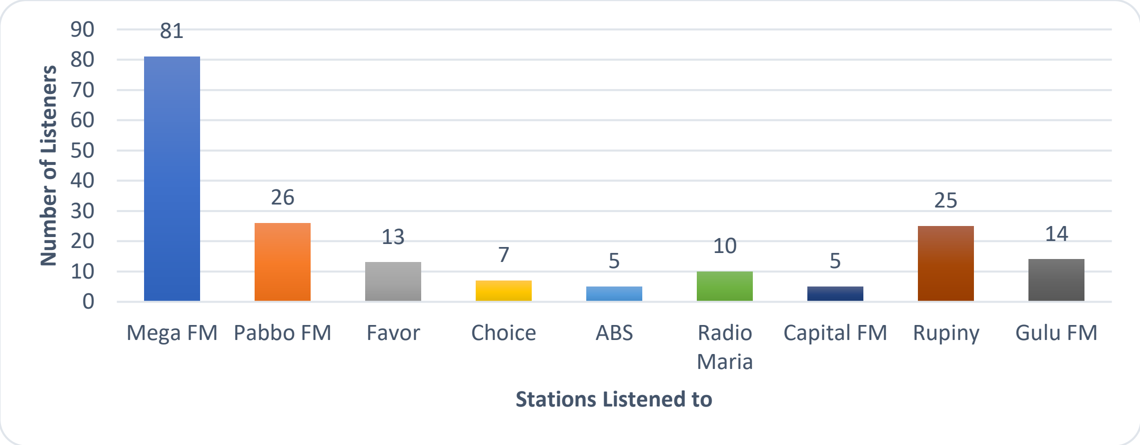



**Figure 4.11: Listenership survey results showed that the community station in Pabbo had climbed to second spot in listenership**

25% of the surveyed participants reported being regular listeners to Pabbo Community FM, the radio station that was set up in the RootIO deployment, with a listenership second only to the state-run radio station. This proves that the station both broadcasts audio of an acceptable quality, and that people appreciate listening to it.

The most persistent request was for an extension of the coverage so that the station covered more areas, beyond the 15-kilometer radius currently covered. However, this goes against the RootIO model as it favors proliferation of small stations, as opposed to growing existing ones to cover larger areas. The licensing authority, however, was not open to the idea of multiple transmitters in close proximity out of fear of interference, though we understand that this stems from their regulation of FM radio at a very large scale. This was a major driver for my work with the SPLAT4J RF mapping tool, to allow communities to visualize areas covered and areas that could benefit from another station.

# Chapter 5

**Contexts of deployment in Europe: Romania**

This chapter reviews the deployment of two community radio stations in Romania facilitated by the RootIO stack. I analyze the performance of the stations in relation to the needs and expectations of the communities, as well as discussing the unique opportunities the stations presented to their communities.

The two RootIO stations deployed in rural Romania were the first ever licensed community radio stations in Romania. Their deployment came after the deployment in Uganda and as such used the platform that had already been developed and built on the lessons learned from the prior deployment. The deployment in Romania therefore helped to evaluate the performance of the RootIO platform, in contrast to the Ugandan deployment, whose outputs were the specifications for the platform. Therefore, the deployments in Romania furnished technical performance data, but also refined the offerings of the RootIO platform based on the experience of users in the community.

The stations set up in two rural communities were licensed under the umbrella community radio organization Radio Civic. Similar to the Ugandan deployment, the stations were supported by a network of media practitioners based in the capital, Bucharest, but working with volunteers on the ground to capture local content and design program formats. Though sharing the essence of the narrative of a grassroots community, the communities in Romania had their own peculiarities that warranted additions to the offerings of the RootIO Platform.

## 5.1 Sfantul Gheorghe

Sfantul Gheorghe is a village situated in the Tulcea county in Romania. It is situated in the delta formed by the southern arm of the Danube River with the Black Sea. Its situation in the Danube delta means the residents are dependent mostly on tourism, given the rich biodiversity that warranted protection as a biosphere reserve. Naturally, the surrounding water bodies also make fishing a prominent economic activity. Other reported economic activities included forestry and small-scale farming. As of 2011 this settlement had a population of 797, with a population density of 1.3 [51] persons per square kilometer, but I observed that the population is concentrated in the main business area.

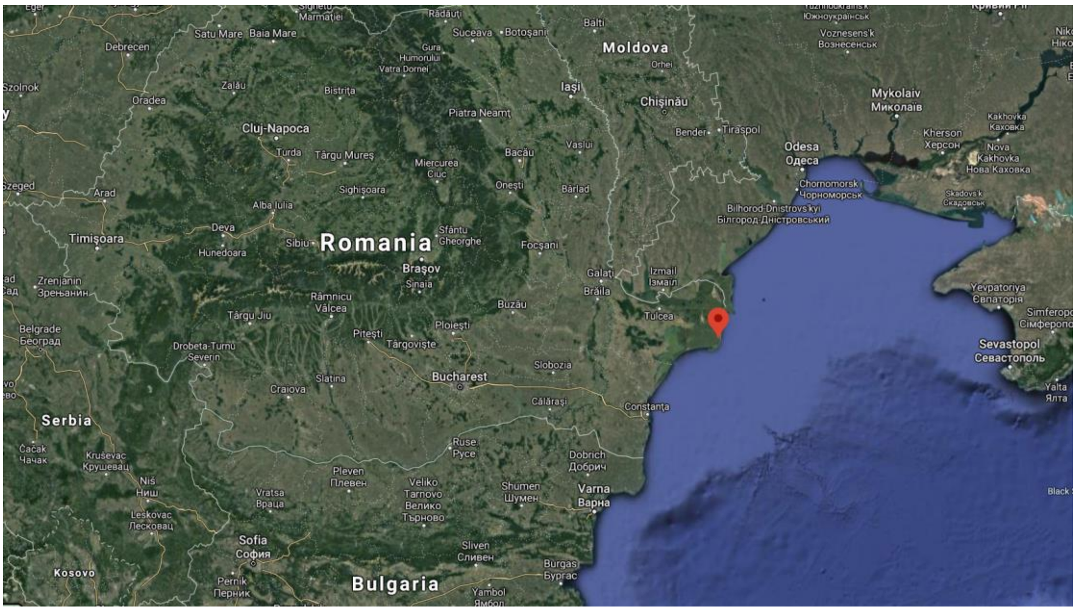



**Figure 5.1: Map showing location of Sfantul Gheorge commune in the Danube Delta**

Given the proximity to Ukraine, the population of Sfantul Gheorghe is largely composed of descendants of Ukrainian origin, referred to as “Hahols” [82]. This estranged, atypical, and shrinking population would not be appealing to many commercial FM radio stations, or even community FM radio stations at the scale that is commonly seen in Europe and parts of sub-Saharan Africa. The deployment of a RootIO station was a perfect example of arriving incrementally at a suitable FM radio deployment tailored to the needs of a particular community.

Sfantul Gheorghe fits the bill of a distanced community mostly in geographical terms: travelling to Sfantul Gheorghe requires a 4-hour drive from Bucharest to Mahmudia, followed by a one-hour journey by speed boat on the Danube River. During my stay, power cuts were frequent, as were mobile network outages, reminiscent of the quality of service witnessed in areas of rural Uganda that shaped the RootIO concept. There was no radio or TV station based in this community, implying that the skills necessary to set up and operate a typical FM radio station would have to be sourced outside the community.

## 5.2 Varvoru de Jos

The second station in Romania was installed in the commune of Varvoru de Jos, situated in the Dolj county of the Oltenia region in Romania. This commune is located 260 kilometers from the capital Bucharest. It consists of 8 villages with a total population of 2,813 inhabitants, a number that is steadily decreasing according to the last population censuses in 2002 and 2011. The total area of this commune is 107.86 square kilometers, making it easily coverable by a RootIO station, which has a coverage radius of 10 kilometers.

A glance at the profile of this community reveals that it is mostly engaged in crop and animal husbandry, attributable to the conducive weather and soil [137]. Varvoru de Jos’ development strategy notes that:

> “sustainable development projects and operational programs work properly when they are part of a defined, assumed, coordinated and applied framework in the interest of the community they represent. The local authority must promote a strategic vision for the future development of the community, which takes into account the interests of the people, the opportunities and the resources at their disposal”

This stance aligns well with ambition behind RootIO; that is, to make FM radio possible within the infrastructural, human and financial resources available within a community. As I show later on, social capital proved key to the success of community FM broadcast, when coupled with other design approaches that enabled communities get to grips with what would usually be an alien technology.

## 5.3 Appropriateness of RootIO

RootIO proved appropriate for the bootstrapping of a community radio station in this context for multiple reasons. First and foremost, the reduced energy requirement meant that the station could depend entirely on solar energy. However, given the capital cost and transportation burden of solar equipment, coupled with the fact that blackouts in this area were not as long and frequent as those witnessed in rural Uganda, it made sense to leverage the electricity grid and rely on batteries for backup energy. The grid energy was also, when available, observed to be of good quality, unlike the brown-outs that I observed in Uganda and that are reported elsewhere by other scholars [75].

Moreover, it is worth noting that while all structures in Sfantul Gheorghe were connected to the grid, in the communities where I worked in Northern Uganda, it was common for most of the structures to not be connected to the nearby electricity grid. This demonstrates how problems of the same category can manifest differently in different distanced contexts.

Another area where the RootIO offering is appropriate for this community regards the costs incurred in setting up and operating the station. A typical FM radio station would have required bulky and expensive equipment whose purchase, transportation, and installation would all have been challenging for this community. In offering a consolidated portable hardware stack, RootIO is able to circumvent all these challenges.

Finally, the absence of media houses in the community would have implied that the operation of an FM radio station based in this community would be extremely challenging. To alleviate this, RootIO offers two advantages: the first is that the platform allows for the automation of sourcing, scheduling, and playing content. The second is that the platform, being cloud-based, allows for the collaborative management of the station. This in Romania allowed experienced media practitioners to assist the communities with cataloging and scheduling content when the community was still building capacity to take on these activities.

## 5.4 Design considerations for Romania

The communities in Romania offered differences that warranted changes to the prior specification of the RootIO software and hardware stacks. While these communities bore the hallmarks of distanced communities, they were significantly different from the Ugandan communities in which I had worked previously, and they allowed for the inclusion of more features in the RootIO stack.

To begin with, as of 2018, Romania boasted a 100% electrification rate, even in rural areas [117]. In stark contrast, the electrification rate in rural Uganda in this period was reported as being 38% [118]. This has several implications for the design of the setup used in Romania. First, the capital expenditure and labor incurred in installing solar energy infrastructure is eliminated, streamlining the process of installing a station. Second, this allowed the use of more powerful 50W FM transmitters. The reach of the transmission is still restricted by the height of the antenna tower, but the signal is much stronger in the area covered, a desirable outcome in the Romanian communes that have forests in the vicinity that contribute significantly to ground clutter which is a major source of signal loss in RF propagation, as explained in chapter 6.7.2. Third, this allowed for the use of more devices such as Raspberry Pi micro computers for recording and streaming, as well as Wi-Fi access points.

Regarding connectivity, Romania is one of the best-connected countries, boasting upwards of 99% LTE coverage in rural areas as of 2019 [38], although from my observation, I noted that not all telecommunication companies offer LTE. Uganda on the other hand had a 65% overall 3G coverage for the same time period [42], this being much lower in the rural areas. I also observed that fixed internet connections in rural Romania were more easily accessible when compared to rural Uganda. This meant that the stations in Romania could benefit more from features such as remote monitoring and streaming content into/from the station. As such, the stations in Romania have more presence on the Internet and are demonstrable examples of the opportunities for low-power FM radio in communities with sufficient Internet coverage.

Even with these clearly encouraging statistics I observed frequent power cuts, despite the existence of the grid, as well as network outages for which there was need for the telecommunications service providers to intervene, often by restarting the networking equipment in the affected region. While the available data verbosely details coverage by Internet and electricity, there is little mention of the use of these two resources. From my experience coverage is often wrongly implied to assume usage, and this can make distanced communities hard to identify when considering only the technological dimension.

## 5.5 Performance of the stations in Romania

Between January 2019 and May 2019, two stations were deployed in the rural communities of Varvoru D Jos in the Oltenia region and Sfantul Gheorghe in the Danube delta. I observed the technical performance of these stations from the start of May 2019 to the end of May 2020. Given that these stations came with support from a cloud portal, increased transmission-site and cloud logging, and synchronization support, it was easier to monitor their technical performance. Quantitative data was therefore obtained by taking metrics from the cloud-based server with which the stations were associated.

I drew more qualitative and quantitative data from surveys conducted by ActiveWatch and MedAlert, two entities that were providing media and technical assistance to these communities to keep the stations operational. A team from these two entities came up with a questionnaires and interviews to construct a lifestyle and information profile for each of the communities. In Varvoru De Jos, 5 people were interviewed and a further 54 aged between 13 and 67 years surveyed with a questionnaire containing 61 questions. In Sfantul Gheorghe, 10 people were interviewed and a survey conducted among 15 respondents aged between 23 and 56 years [106]. More information was also collected by the MedAlert and ActiveWatch teams during field visits totaling approximately 190 days in both communities [106], and ad-hoc communication with members of the community.

### 5.5.1 Scheduling and programs

During the period in which I observed the two stations, a total of 36,452 programs were broadcast on both stations. The majority of the programs were cloud programs (86%), meaning that they were managed and orchestrated by the cloud platform, while the remaining programs were delegated to the device at the transmission site. For this section, I will refer to the programs orchestrated at the transmission site as “local programs”:

| **Station** | **Program Type** | **Number of broadcasts** |
|---|---|---|
| Varvoru de Jos | Transmission site | 2,741 |
| | Cloud | 15,082 |
| Sfantul Gheorge | Transmission site | 2,313 |
| | Cloud | 16,316 |

**Table 5.1: Breakdown of programs across the stations in Romania**

While it may seem that there were few programs local to the transmission site, this is because they were much longer (spanning four to six hours). They were used to provide a background of audio content, mostly music from the device’s local storage, that covered the station’s entire daily schedule with cloud-based programs overlayed at desired times. As the user will recall, cloud-based programs have precedence over, and can interrupt, local programs. This setup worked well when the cloud-based programs could not be initiated, providing backup media for broadcast.

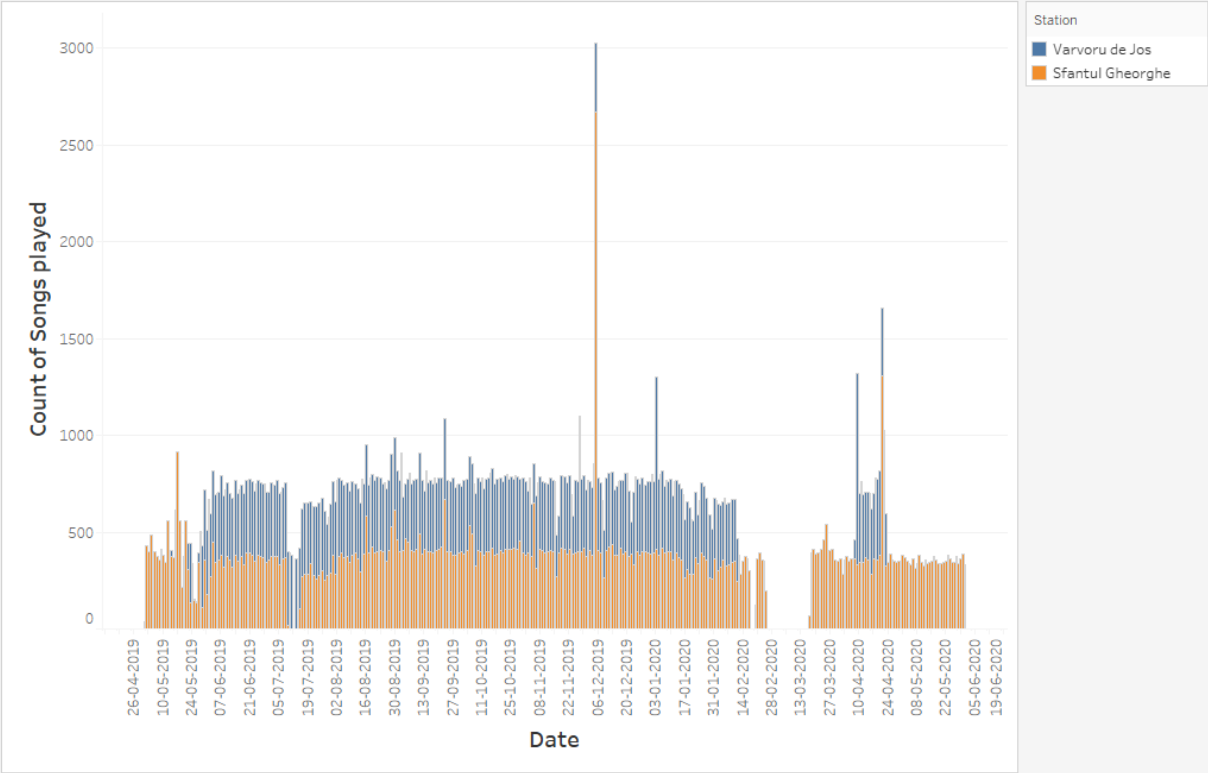



**Figure 5.2: Graphic showing the number of songs played per day in each of the Romania stations**

On average the local programs were playing just below 400 songs every day on each of the stations, as can be seen in Figure 5.2 interspersed with roughly four streaming programs on the station in Varvoru de Jos, which as I show later, had better connectivity compared to the station in Sfantul Gheorghe. As can be seen from Figure 5.9, more streaming programs were subsequently incorporated into both stations, showing a shift in preference to streaming audio content as opposed to preloading it to the phone, which would have stifled dynamic content programming.

Initially, programs orchestrated from the cloud were shown as yellow on the schedule, while programs playing locally were displayed with a blue color on the schedule. Following complaints that some programs were not playing at the time that they were scheduled to play, it was deemed necessary to log the status of programs that were aired, and optionally send out a summary log, as well as display the status on the calendar with red representing failed programs and green representing successful programs, as can be seen in Figure 5.3.

```
19-09-11 12:00:05 Done initing Media action for program 132205
19-09-11 12:00:07 Loaded playable media
19-09-11 12:00:17 result of station call via SIP is (True, '8e67db46-48a5-4aed-82e2-937866997a4f\n')
19-09-11 12:00:17 Received call answer notification for Media action of 132205 program
19-09-11 12:00:17 Playing media intro_pilula_de_sanatate.mp3
19-09-11 12:00:18 result of play is +OK Success

19-09-11 12:00:23 continuous play is on, will move on to the rest of the episodes (28)
19-09-11 12:00:23 will now play episode #2
19-09-11 12:00:24 Loaded playable media
19-09-11 12:00:24 Received call answer notification for Media action of 132205 program
19-09-11 12:00:24 Playing media Pielea.mp3
19-09-11 12:00:24 result of play is +OK Success

19-09-11 12:00:24 result of station call via SIP is (True, u'8e67db46-48a5-4aed-82e2-937866997a4f')
19-09-11 12:03:18 continuous play is on, will move on to the rest of the episodes (28)
19-09-11 12:03:18 No more files to play. Should proceed to hangup
19-09-11 12:03:18 Deregistered, all good, about to order hangup for 132205
19-09-11 12:03:18 result of stop play is -ERR Cannot stop displace session!
```

**Figure 5.3: Excerpt of a program summary sent out after a program aired with information about different actions executed during the program**

From the week of the 18th of April to the week of the 16th of September 2019 the two stations were processing between 500 and 700 cloud-based programs per week. As can be seen in Figure 5.5, the success rate ranged from 23% at the beginning of this period to 85%, before an examination revealed that for some of these scheduled programs, media was never uploaded to the server and they therefore were not played at their scheduled time, but rather labeled as failed. Programs of this nature were highlighted on the schedule (colored grey), and in addition they were separately categorized once the scheduler skipped them. This allowed the communities to clean out the programs that were no longer needed or for which content could no longer be sourced.

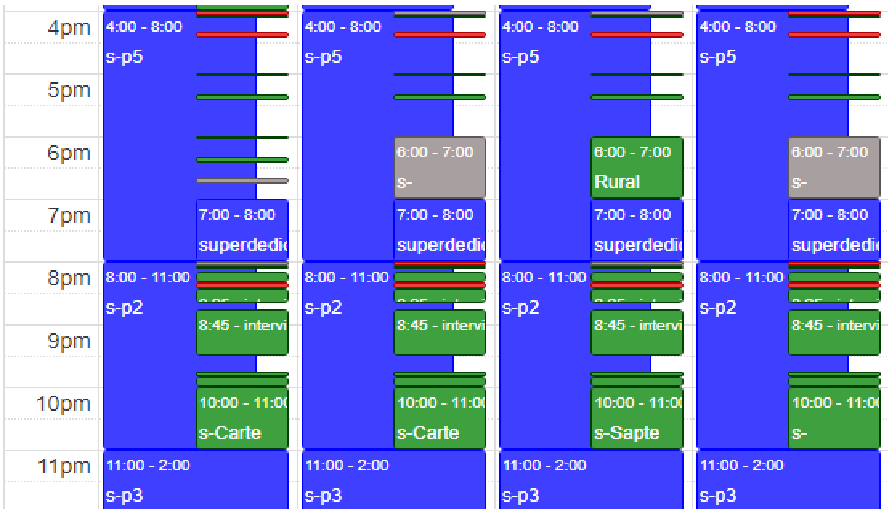



**Figure 5.4: Section of schedule showing local programs (blue), failed programs (red), successful programs (green) and programs with no associated media (green).**

From the week of the 23rd September 2019, higher success rates were recorded as can be seen in Figure 5.5, and this increased confidence in the system. This was a sharp reminder that all of the work that had gone into the system could have gone to waste if it did not align with community expectations, made unfounded assumptions about community capabilities, or left some scenarios open to interpretation by communities. To address these issues, a Wiki (http://wiki.rootio.net) was created that explained in detail to the communities the different available functions, and the option that should be implemented where different design options were available.

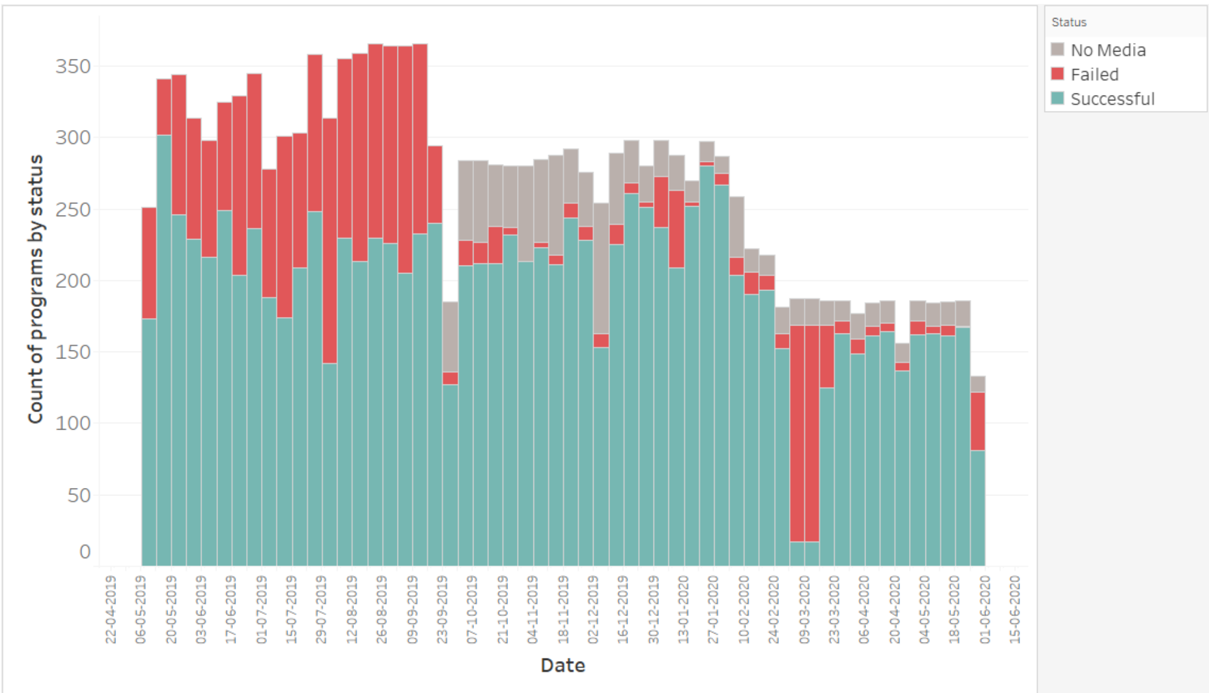



**Figure 5.5: Graphic showing cloud programs aired at the Sfantul Gheorghe station**

## 5.5.2 Internet use and availability at the stations

My initial interactions with members of the community and measurements revealed that there was sufficient internet coverage, even in remote regions of Romania, to warrant exploration of a design that accommodated more Internet offerings. The two stations exhibited different network availabilities, and the effects of the network type, though visible, were not significant to the performance of the stations.

Measurements of the Internet speed were taken every time that the stations synchronized with the cloud server, during the routine requests in which they uploaded analytic information about networking, power/battery capacity, storage utilization, memory utilization and Wi-Fi connectivity. Accessing similar endpoints with similar upload and download payloads, the station in Varvoru de Jos averaged a synchronization duration of 445.49ms, while the station in Sfantul Gheorghe averaged 713.85ms.

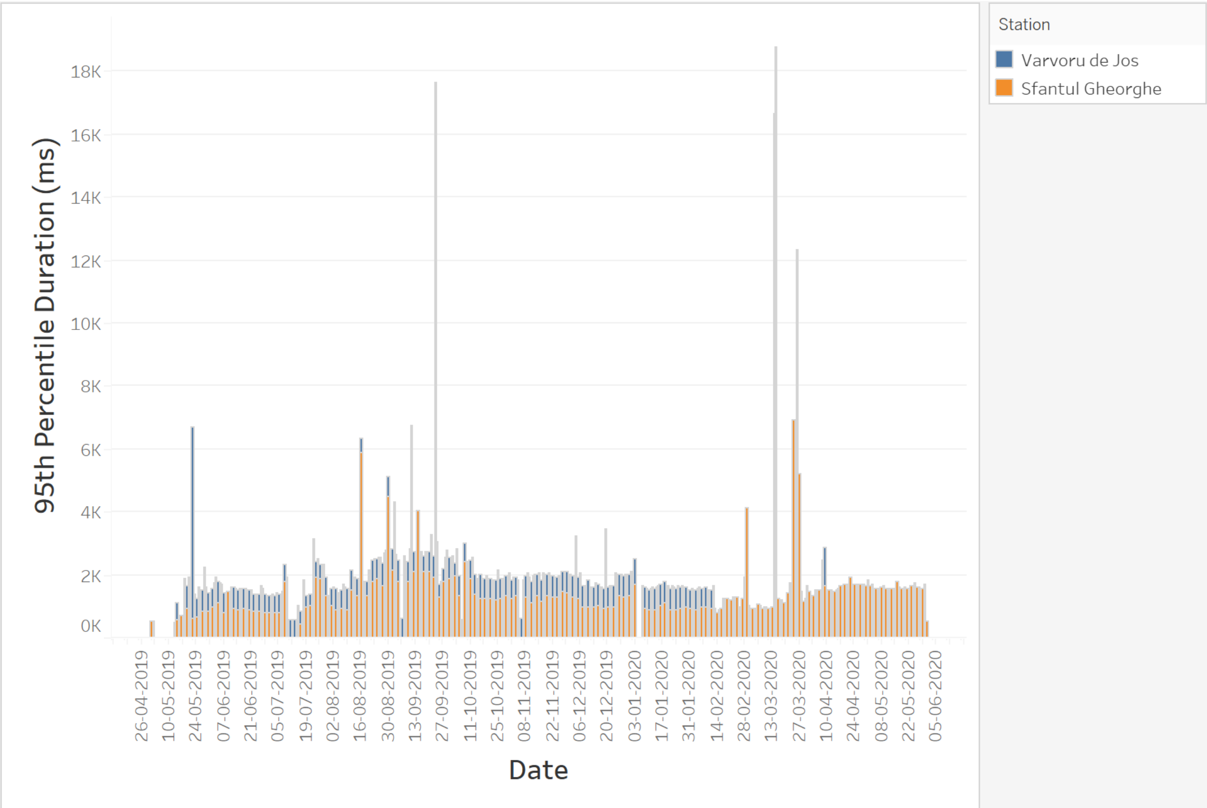



**Figure 5.6: Graphic showing the daily 95th percentile sync durations in the two Romania stations**

The difference can be explained after an examination of the network types that were being used by each of these two stations. The station in Varvoru de Jos was connected to a 4G LTE network for 70.8 % of the time, and 18% of the time to a 3.5G network. On the other hand, the station in Sfantul Gheorghe was mostly connected (41% of the time) to a 3.5G network, and 36.1% of the time to a 3G network. An inspection of the usage trends of the two stations, however, revealed that the difference in speeds was not an issue given the low-Internet use that is central to RootIO's design. Overall the 300ms bin was the modal bin for both stations, though the 90th percentile can be seen at the 500ms bin for the Varvoru de Jos station and at the 900ms bin (Figure 5.7) for the station in Sfantul Gheorghe (Figure 5.8). This implied that the network in Sfantul Gheorghe was usable but subject to frequent degradation.

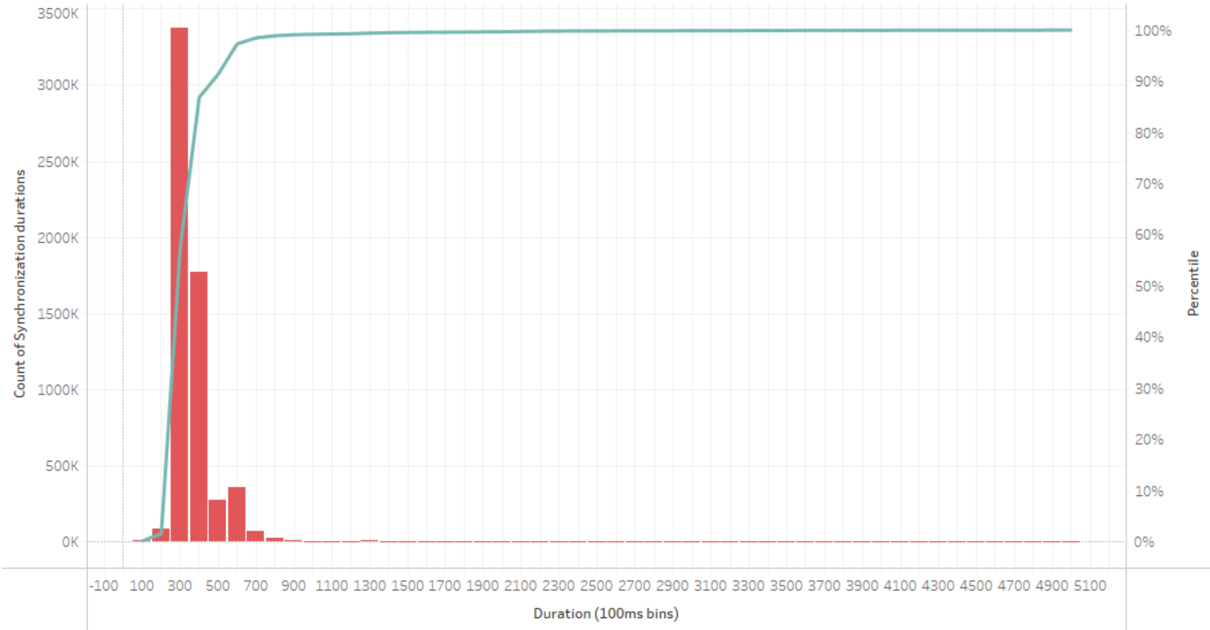



**Figure 5.7: Distribution of durations of synchronization requests on the station in Varvoru de Jos**

The decision to store media files locally on the phone, with the ability to organize them into playlists, meant that the station did not have to depend on the Internet for almost all the high-quality audio content. Only short, frequently changing media, such as news bulletins, were played through cloud programs,

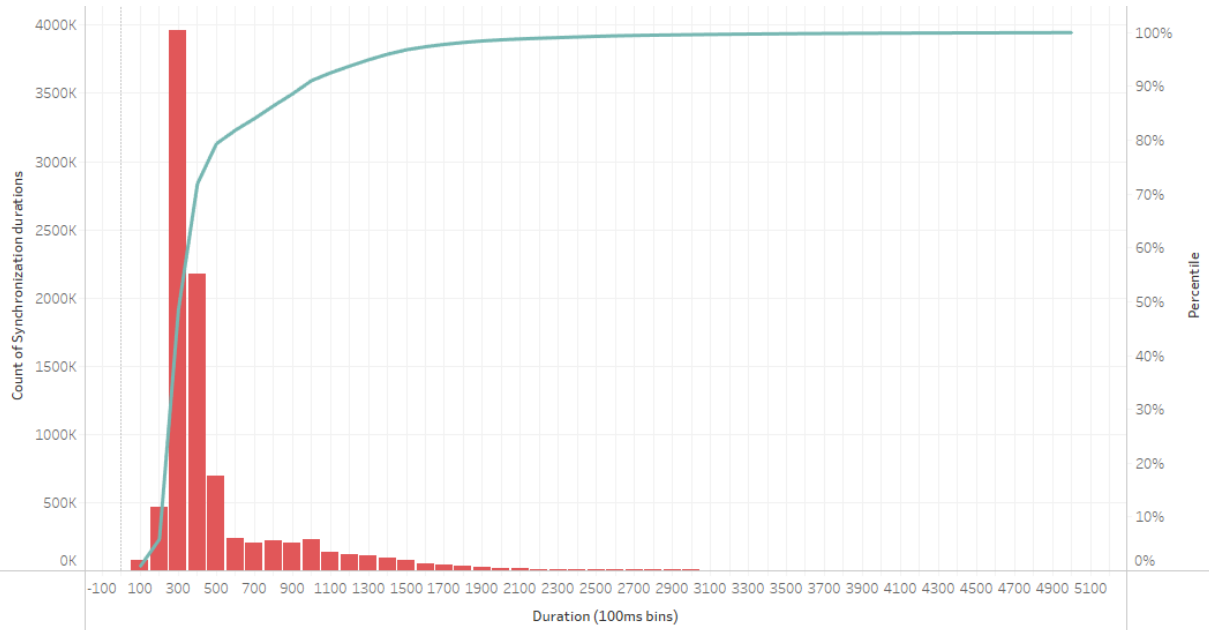



**Figure 5.8: Distribution of durations of synchronization requests on the station in Sfantul Gheorghe**

and the fact that these mostly consisted of spoken content made it possible to leverage low-bandwidth telephony with little perceived loss in quality.

While the option to stream the files over HTTP was technically available, the preferred approach was to stick to a telephony model in which the calls were streamed through a call made between the cloud and the transmission site. The advantage of this approach was that when a high-speed Internet connection was available, the call made through SIP could use higher quality audio codecs such as Opus, but could fall back to normal voice calls via PSTN networks when Internet was not available, with no changes necessary in the cloud software logic.

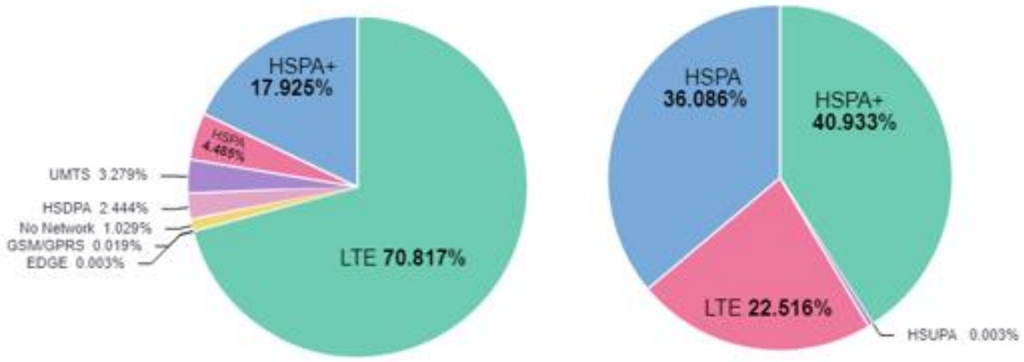



**Figure 5.9: Graphic showing network access by technology for the stations in Varvoru de Jos (left) and Sfantul Gheorghe (right)**

## 5.5.3 Community innovations around the stations

As previously mentioned, the RootIO setup that was offered to communities in Romania was shaped by experiences in Uganda and was therefore barely dependent on the availability of Internet at the transmission site and still did not consider the possibility of even the most basic studio setup at the transmission site. However, as RootIO allowed these communities to interact with a technology from which they were always far removed, very soon the communities were able to craft their own additions and customizations to the RootIO platform.

Central to these additions was the increased sustainable access to relatively fast Internet. The functions facilitated by the initial RootIO offering, such as streaming audio content over phone calls, were very rudimentary in the face of high bandwidth Internet. I made enhancements to facilitate increased access to high quality media through the Internet. In the RootIO architecture this involved the addition of stream reception capabilities at the transmission site and the ability to create and test streams in the web portal that is used to manage the stations. News files and other media that were previously played via SIP and PSTN calls were subsequently played into the station through media streams over HTTP.

Another area where the RootIO solution offered a rather rudimentary solution was in the conduction of talk shows where the implementation involved making a conference call to the host of the talk show, with the ability for other members of the community to call into the talk show by dialing into the conference call as detailed in chapter 4.2.2. This approach had proved hard to realize given the steep learning curve for the host, the amount of telephony involved and the eccentricity of the process. The preferred solution was to arrange for a studio-like setting with audio fed into a computer, which the


**Figure 5.10: Snippet of Radio Civic website showing Stream players for the Romania stations**

computer fed in turn into a stream that was picked up by the transmission site. The same approach was used when individuals had spare time and wanted to deejay on the stations.

Another area that was facilitated by increased access to the Internet was outbound streaming for the stations in Romania. Given that the two communities I worked in were some of the most remote in Romania, many of their members had emigrated to more urban settings, and the only plausible way for the station to reach them was via the Internet. As such, the communities set up a streaming service where the audio bound for the transmitter was also fed into a cheap Raspberry Pi computer, and broadcast through publicly accessible Icecast streams linked to on the website of the two stations, as can be seen in Figure 5.10. This aligns with my thesis that initiatives in the FM radio can bootstrap an electronic communication culture with presence in digital spaces.

The streaming microcomputer also recorded the content that was being played, allowing for content archival: a key requirement of the regulation in Romania pertaining to the operation of FM radio stations. The technology developed for these functions was contributed to the RootIO repository, and is part of the standard plugins maintained in the RootIO project.

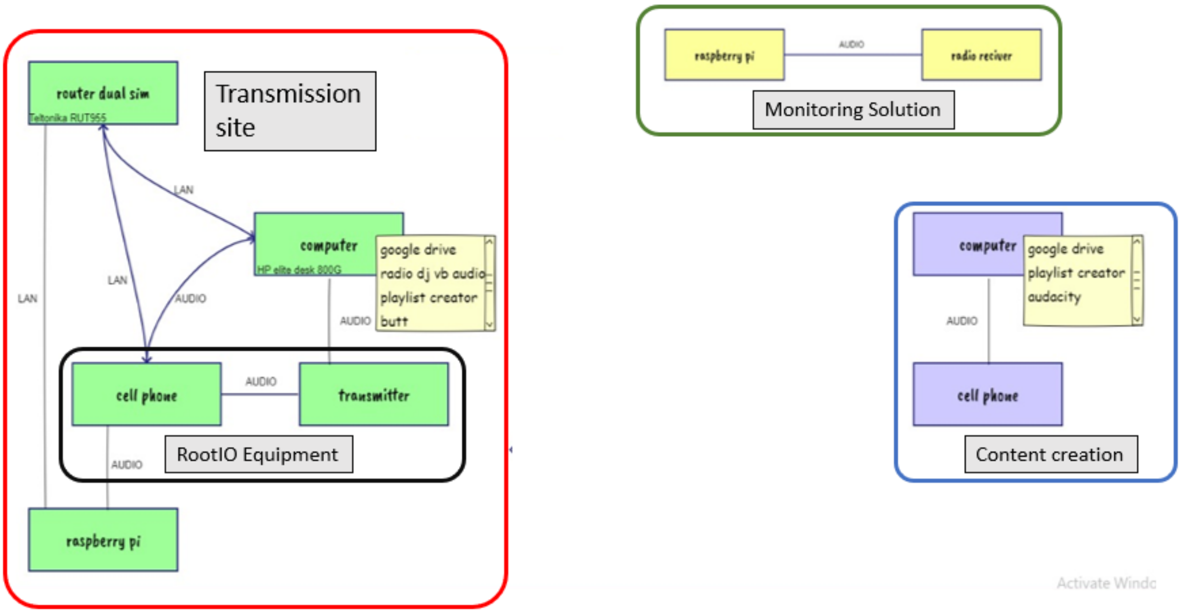



**Figure 5.11: Graphic showing the eventual technical composition of the RootIO stations in Romania**

# 5.6 Lessons and outcomes

The RootIO offering enabled the creation of the first two Romanian community radio stations. The initial offering was basic and below what was possible within the constraints of the communities. The simplicity of the solution however allowed communities to consider this technology, examine the interaction between the basic components necessary for FM broadcast and craft additions that would make the technology more relevant to their settings.

By observing the communities in Romania interacting with the two RootIO stations, participating in discussions with volunteers and organizations supporting the stations, it was possible to learn how best to situate FM radio in distanced regions, as well as how to support and consolidate community contributions. Below I elaborate the lessons from RootIO that would benefit practitioners seeking to proliferate communication platforms in distanced regions.

## 5.6.1 The phone as a less-than-ideal platform

Among the new lessons learnt were the numerous shortcomings of using the smartphone as a computing platform for long-term computing. Given the increased load on the phone, in terms of the number of media files played and the number of phone calls received, both on the SIP endpoint and the native calling application, it became clear that the phone was being used in a manner that contrasted starkly with the one for which it was optimized.

Smartphones are increasingly being designed to minimize battery usage, limit access to the Internet, which can be expensive, and protect the privacy of their users. The RootIO application for Android was frequently using the Internet, almost always running the Internet or streaming phone calls, and required permissions to read files on the phone, read SMS messages, and, in some cases, even access the microphone. The phone was therefore always working against the RootIO app by trying to limit connectivity, and by trying to send the application to the background where it could be put on sleep mode.

In some intermediate Android versions, it was even not possible to programmatically answer phone calls, a feature that was central to RootIO's design. Some of the features that were used by RootIO also necessitated "rooting" the phone for the app to have permissions beyond those permitted by the OS in an off-the-shelf phone. This was challenging and limited the ability of communities to carry out deployments without the direct intervention of the RootIO team. Lastly, given the many permissions that the RootIO app needed, it was barred from the Android app store, making updates and installations challenging.

As such, it was determined that the smartphone, though offering the benefits of a well-integrated media player with telephony and programmability, was not the right platform for a long-running solution like the one RootIO deployed at the transmission sites, even though the loads on the phone were reasonable according to profiling experiments. Chapter 6 is dedicated to assessing the performance of the Raspberry Pi, which was identified as the most appropriate replacement for the phone at the transmission site.

### 5.6.2 Plug-in vs extension

The idea behind RootIO was the design of an open and extensible stack that communities could leverage to set up and operate low-power FM radio stations. The aim was for communities to take the code base and customize it in ways that suited their local settings. However, I noticed that in the case of the Romanian stations, this would have introduced functionalities that are not relevant or usable in other contexts and would also offer a steep learning curve given the size of the code base.

As such, I opted for a design that allowed for communities to develop basic plug-in systems that could be integrated with the RootIO platform through common standard protocols and approaches. This is an approach common in the open-source world, and relied on by initiatives such as Linux. The Romania deployment alone saw a range of interfaces introduced to the RootIO platform:

- A TTS engine for generation of spoken content derived from text sources.
- Content repositories exposed through RSS feeds.
- An implementation for audio streaming and content recording for regulatory compliance purposes.
- An option for remote monitoring and configuration of computing devices at the transmission

site.

The impact of this was significant because it allowed the community greater flexibility in terms of content that could be injected into the station as well as addressing regulatory concerns. As such, the standard preference now is to maintain the core of the RootIO stack as a few services that can guarantee the minimal broadcast requirements of any regulatory context, while allowing communities to integrate their own sustainable improvements as well as pick from a growing list of add-ons contributed by other communities. This, too, is in line with my philosophy of additively constructing community radio as opposed to subtracting from commercial broadcast radio.

### 5.6.3 Social impact of the station in Sfantul Gheorghe

In Romania, while the station in Sfantul Gheorghe had worse electricity and Internet conditions, its performance and engagement were much higher than that of the station in Varvoru de Jos. By comparison, the station in Sfantul Gheorghe is in a far more remote region and therefore serves a more distanced population. The station however was central to the information culture within the community. Even the prominent informational social media pages were formed around the station, giving a clear example of what I envisaged when I theorized that FM radio, though seemingly archaic technology, has the potential to kickstart an electronic information culture in distanced communities.

In a survey conducted among 45 members of the community, only 2 members reported not listening to the station. Of the listeners, 65% identified the RootIO station as their first choice, with another 14% reporting the RootIO station as their second choice of station in a total of 7 stations that are accessible in the area. This was similar to the trend observed in Northern Uganda (chapter 4.4.3) where the local station quickly gained listenership over existing larger stations.

As of March 2021, the station in Sfantul Gheorghe featured 15 programs crafted within the community and presented by 20 local volunteers. In line with my thesis, the listeners of the station in Sfantul Gheorghe demonstrated differing taste in music and language from the general Romanian population, taking for example the music chart [138]. Given that the population in Sfantul Gheorghe is predominantly "Hahols" of Ukrainian descent, programs that promoted the local culture were appreciated. Listeners showed a substantial preference for Ukrainian music (35%) and Russian music (19%).

The community clearly had access to music and information from outside the community – favorite artists mentioned included global icons such as Pink Floyd and The Beatles – but the most preferred content on the station included programs about locals and traditions (33% of respondents), local music (30% of respondents), local news (26% of respondents), and interviews by a local volunteer (21 % of respondents). This goes to show the abundant space for grassroots FM radio that is currently available, even within communities that have access to multiple commercial FM radio stations.

A key theory in my thesis is that ownership of FM radio at grassroots level is capable of organically growing an electronic information culture aligned with the values of a particular community. This was evidenced in Sfantul Gheorghe where patterns prevalent in online platforms started to emerge in

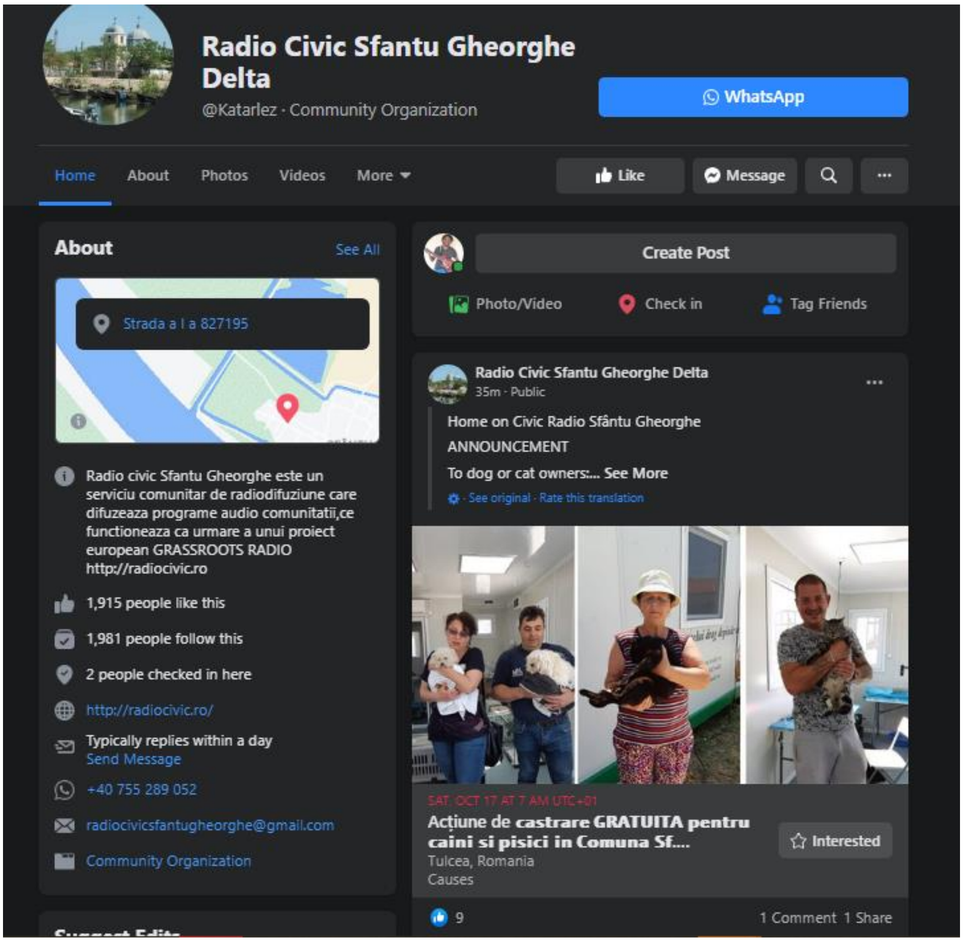



**Figure 5.12: The Facebook page of the community radio station in Sfantul Gheorghe**

the community. A Facebook page was set up for the radio station, which complemented the analog broadcasts, giving rise to an ecosystem that comprised of both analog and digital platforms. While it is always easy to resort to the Internet for most solutions that involve communication, it is possible to organically conceive ideas around existing technologies when the skills and infrastructure necessary to harness the Internet are in short supply. In Sfantul Gheorghe, members of the community called in their orders to have merchandise delivered to their homes after shops placed informational ads on the station, akin to prevailing e-commerce facilitated by the Internet. This is a community for which e-commerce as we know it would not be practical, but also for whom contemporary FM radio is not suited. This is an example of a community organically leveraging a technology that they understand, as opposed to a model in which designers would be quick to design web-based e-commerce approaches.

Discussion with members of the Sfantul Gheorghe commune revealed a very localized and closely tailored model of FM radio. One resident explained:

> "The news from the village. During the winter, you don't see each other and that's how you hear something new on Radio Civic. In the summer we are busier, and we meet each other, and we find out."

It is not typical to think of commercial FM radio as a medium that facilitates individual communication within communities given its size in relation to this role, and the emergence of Internet-based social media, which has provided a more appropriate space for this kind of interaction. One of the members noted that information that was always posted on the notice boards of the city hall and risked being missed if people stayed home, found its place on the local radio. Again, this was mostly possible because the station was appropriately sized in relation to the community.

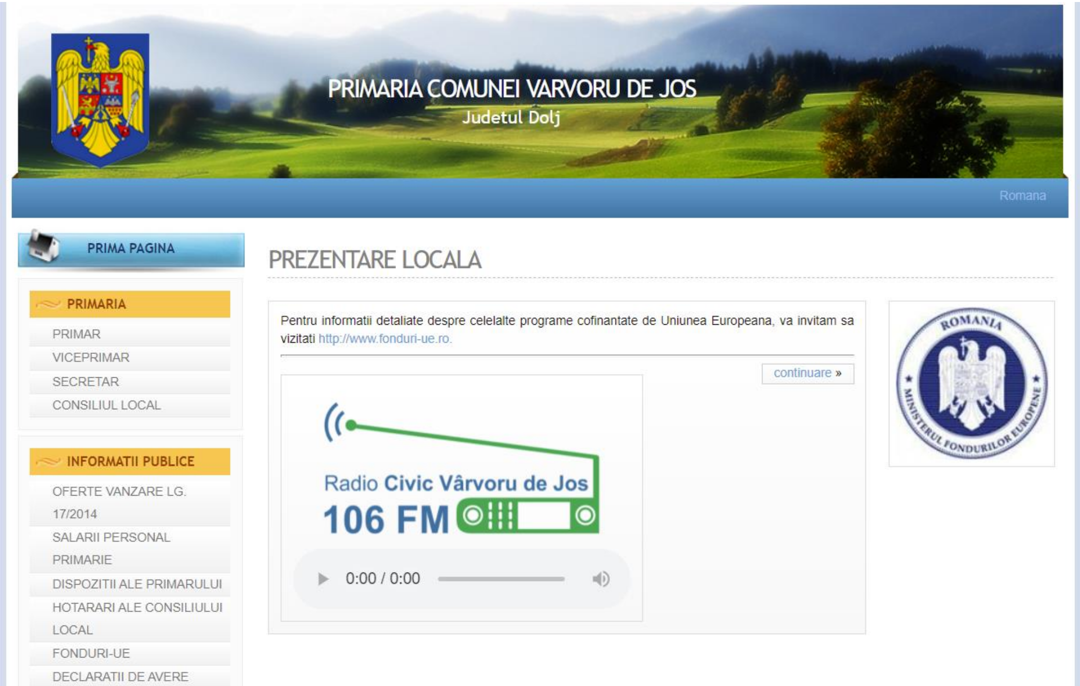



**Figure 5.13: Snippet from the home page of the site of the Varvoru de Jos Council with an embedded player for the radio station**

When it came to the appreciation of the station's service, it was identified that most users appreciated the service it was doing towards providing local information (33% of respondents) and connections and culture (49% of respondents). The station also became the natural residence for information about economic drivers in the commune such as fishing, electricity schedules, and notices regarding the ferry service. Community members identified advantages that served the unique interests of Sfantul Gheorghe, which is comprised of an ageing population with roots in another country.

The station in Sfantul Gheorghe proved some of the core theories espoused in my thesis. The first is that communal involvement is critical to the sustainable design of communication solutions in distanced communities. Despite being deployed in relatively more adverse conditions, this station performed much better given the community involvement and sense of ownership. A second theory is that ownership and control over a communication technology that people are familiar with would result in bootstrapping an electronic, even digital, information culture, that is unique to particular distanced communities. This was proven by the station becoming the nucleus for the community communication strategy including via digital channels such as web and social media pages.

Perhaps the most crucial theory is that incrementally constructing FM radio and taking care to include only the necessary components would result in a more appreciable community radio. In differing from a model that subtracts from mainstream FM radio to arrive at community FM radio, this model carefully examines the components that are added, evaluating them for relevance to the served community. This is fundamental for the achievement of sustainable and relevant community FM radio.

# Chapter 6

## Consolidating the technical lessons

In this chapter, I take the lessons learnt from deployments in Uganda and Romania and use them to formulate an optimal technological design, whose performance I evaluate in a controlled environment. I also discuss general high availability considerations incorporated into the design of the RootIO stack.

It became clear, from observing the stations in Romania, that the opportunities to make changes that would make the most significant impact lay in the areas accommodating the software which operated RootIO stations in the cloud and at transmission sites. The other aspects of the RootIO platform such as the transmitters, towers, and antennae in use, all came down to logistical possibilities that are outside the scope of my thesis. That said, I did participate in the design of DIY antennae [88] using easily available materials, some of which were tested in Northern Uganda, yielding an SWR of 1 and capable of covering a 12km radius. The need to import RF cable and transmitters, however, made this work redundant given that it did not solve the most pressing logistical issues in this area.

During my research I also worked with sociologists and Human Computer Interaction (HCI) experts who studied other less technical components of RootIO stations such as the impact of composition of management boards and ownership structures, interaction with regulators [11,21], optimization of content generation [93–96], and democratization of the FM space utilized by RootIO [19,20]. There is a wealth of literature generated from studying various social propositions of RootIO and the model of FM radio, in general, that it proposes. I paid close attention to this research, and it shaped the later improvements to the RootIO platform.

My original design for RootIO hardware at the transmission site was geared at evaluating broadcast feasibility in remote regions characterized by scant electricity and telecommunication networks. With this in mind, Android mobile phones were an ideal platform for a prototype, but soon became restrictive once deployments transitioned from proving the concept to actually deploying stations with the intent of using them for regular broadcast.

Firstly, I recap the strong points, but then go on to highlight the features of Android phones that made them restrictive and warranted the move to other computing platforms capable of utilizing the same or equivalent advantages as the mobile phone. I then detail the architecture of the RootIO app on Raspberry Pi and conclude with an analysis of data showing how the additions made, and approaches undertaken, demonstrating improved performance during evaluation, and therefore evidencing the gains over what was possible using the mobile phone. Lastly, I go through the performance optimizations on the cloud platform and discuss how they address issues that were observed during the operation of RootIO stations in Romania.

## 6.1 Strengths of Android mobile phones

Given the challenges witnessed using the Android phone, and the space in this thesis devoted to addressing them, one might conclude that I created the problem for which I spent so much time seeking a solution. The use of the mobile phone, however, allowed for faster deployment of a prototype, and allowed for an end-to-end examination of potential solutions for low power broadcast. Questions around how to leverage telephony for community inclusion, interweaving dynamic content with relatively static content, and possible applications of intelligence at the transmission site, were made simple to explore by using the mobile phone as an initial prototyping platform. The result is that the latter technological stack, based on a Raspberry Pi computer, is the fruit of a design that is already tailored to grassroots communities, which is a big win given the greater outlay of effort needed to set it up. Below, I elucidate factors that influenced the choice of the mobile phone as an initial prototyping platform.

### 6.1.1 Ubiquity

Mobile phones are widely available and are familiar technology, even in distanced regions. This is due to their portability, use of wireless networks, and possession of batteries that make them usable even in areas with little to no electricity grids. As such, they are ideal prototyping platform for a DIY technology project that involves electronic communication. This also means that, even in rural communities, there is sufficient knowledge of the basic operations involving mobile phones, such as connecting to Internet networks, transferring media files, and installing apps. These skills were very handy when working with communities to get RootIO stations going.

Among smartphones, Android OS is the most prevalent, with a market share of 71.9% as of February 2021 [139]. This is with good reason: by decoupling the OS from actual devices, Android benefits from a variety of hardware platforms furnished by OEMs, targeting different geographic and economic market segments. The open-source nature of Android also allows for contribution from tech enthusiasts as well as for the creation of custom builds by device manufacturers and individuals. The resulting effect is an Android device for almost every demographic and in any dimension.

### 6.1.2 An almost full JVM and richer computing platform

Android's Dalvik Virtual Machine and subsequent Android Run Time support almost full specifications of the Java Standard Edition language specification. Unlike in earlier mobile phone computing platforms that supported the minimal J2ME standard, the Android platform provides many more language options and control over the mobile phone in most of the areas used by the RootIO application, such as media play, networking, telephony, and file manipulation.

In addition, Android platforms are built on a Linux kernel, giving a much more versatile and fuller computing platform than what was previously available in mobile phones. This means that, among other things, there is support for common tools such as an SQL database, which was crucial for RootIO's operation. Internally, Android stores call records, media libraries, and SMS messages in an SQLite database that makes for easy querying by the RootIO app. The implementation of entire SIP telephony stacks to facilitate IP telephony was also available, something that would be hard to achieve with phones supporting only the J2ME standard. The Android API provides a decent implementation of a media player, but it was also possible to import a library that offered even more robust media play capabilities.

### 6.1.3 Solid Network Integration (Wi-Fi, SIM, GPS)

Mobile phones have very good telephony integration, given that this is what they are primarily designed for. Attempts to add telephony to computers using USB modems, GOIP devices, and add-on HATs all require significant configuration to get the interaction working, but also require significant code to control and communicate with the telephony device over serial ports.

Android, on the other hand, exposes the telephony service via APIs that allow for retrieval of network status as well as manipulating phone calls, as was required by the RootIO app. The phone is responsible for maintaining the inbuilt telephony modem in an operational state and notifying registered listening applications of any changes in the status of the telephony modem. An added advantage was the legitimization of programmatic answering of phone calls through PSTN networks starting with Android version 8. Previously, this was possible only through unofficial hacks that involved triggering an event that simulated the pressing of the answer button on a wired headset, a loophole that was fixed with Android version 5, hence rendering it impossible to use devices on Android version 5, 6 and 7 for RootIO purposes.

Given the added communication requirements of modern mobile phones, extra connectivity options are added, which I utilized in the RootIO application. Mobile data connectivity allowed for synchronization with the RootIO cloud in areas where Wi-Fi networks were non-existent, but, in the areas where Wi-Fi networks were available, phones were observed to easily switch between Wi-Fi and mobile data, depending on which technology provided connectivity. Lastly, Android mobile phones have GPS connectivity, allowing for key features such as shutting down all RootIO services once the phone is deemed to be too far from the licensed broadcasting location. By extension, this can also be used to control the transmitter, and to shut it down if it is in an unlicensed location.

### 6.1.4 Rich features and media options.

Android mobile phones offer an increasingly rich set of features that I utilized for the RootIO project. To begin with, Android phones have support for large expansion storage cards, accommodating tens of thousands of music files at the size typical of MP3 format. It was possible to enrich the existing collection by ejecting the media card from the phone and adding more songs to it while it is inserted into a computer. Optionally, it was also possible to add media by plugging a USB cable from the phone into a computer to copy songs, beam song files via Bluetooth or even, where supported, connect a USB flash disk to the phone using an adapter to copy songs over. As will be seen in chapter 6.4.3, these functions are not trivial in relation the microcomputer option that I chose to replace the mobile phone, but plausible alternatives offer a potentially more robust and fail-proof solution.

The increased media sourcing options allowed for a local music repository at the transmission sites of the RootIO stations. In addition, the Android SDK includes an API for media play that supports simple formats and functions. However, given the variety of audio formats that I observed in these communities, coupled with the observed instability of Internet connections, I opted to rely on Exoplayer (https://developer.android.com/guide/topics/media/exoplayer): a third-party media play library with support for more audio formats, better streaming capabilities, and more granular events related to state changes during media play.

The Android platform also maintains a catalog of all media files on the mobile phone, allowing access to them either via a content provider API or by direct access to the SQL database housing the records. This catalog is updated every time media files are added to a certain directory or when the SD card is (re)inserted. This function was critical for RootIO as it allowed the creation of playlists in the cloud that could be mapped to media on the phone and associated with programs. An advantage here was that irrespective of where songs were stored on the phone, it was possible to retrieve them by querying the media catalog for their location using the title, artist, or album as a key.

### 6.1.5 Great messaging and trigger options

The Android platform offers great options for event passing and handling. These proved critical for executing remote functions on the phone to manipulate the RootIO app on the phone. Android allows for a particular app to register listeners for “intents” matching a certain filter. In my case, I used this function to allow the RootIO app to listen for SMS messages that were being sent into the phone, and for executing some actions in the RootIO app if the message payload matched that actionable by the app. An advantage of this approach was that this action could be registered to be executed even when the RootIO app was not running, and could thus be used as an entry point to start the RootIO app in cases where it was killed irrecoverably by the phone OS. As we will see, this was crucial given that Android always tries to suspend long-running background applications.

The same “intents” functionality can also be used for asynchronous communication within Android apps, but also between the app and other components exposed by the Android SDK. Events related to the phone modem, such as changes in call status and incoming SMS, changes of location, and changes of battery status, can all be subscribed to, and wired into, the flow of the RootIO app. The same functionality facilitated listening to Internet connectivity changes, as well as listening to scheduled alerts, signaling the start and end of programs. As I will describe later, recreating this message passing functionality was not easy on the Raspberry Pi computers that I use for my final design.

### 6.1.6 Cost of ownership and operation

While smartphones have always been considerably more expensive than feature phones, Android, by being hardware-agnostic, has greatly contributed to the lowering of smartphone prices. On other hand, iOS, the second most prevalent phone OS, is hosted on devices priced in a narrow but significantly higher price band. It is therefore no surprise that Android dominates the smartphone market. The devices that were used in RootIO deployments ranged in price from €50 to €250, but the real advantage here is that members of these communities could afford these devices prior to the deployments and were therefore already familiar with routine operations such as transferring files and installing applications.

Mobile phones, similar to portable FM radios, thrive in areas that have scant grid electricity infrastructure. For example, in a community like Pabbo that relied mostly on energy supplied by solar-charged batteries, mobile phones are fully useable, while Raspberry Pi computers may require some configuration functions, which in turn necessitate a computer screen that runs on AC power, for which an inverter would have to be added to the setup.

### 6.1.7 Extra features afforded by rooting phones

Android phones restrict some actions that would otherwise be dangerous to owners of mobile phones, if those functions were accessible to third party applications. The process of rooting involves unlocking the bootloader of the mobile phone to allow installation of a version of the OS in which critical functions are made accessible to third party applications. One function that is facilitated by rooting the phone is the ability to reboot the mobile phone programmatically. This function coupled with the "intents" messaging function meant that phones can be remotely rebooted by sending an SMS.

Another function possible once the phone is rooted is the ability to remotely access the screen of the mobile phone and control it. This function allowed users to check on the phone as well as verify and define various settings. It also was possible to log in remotely and re-install, upgrade, or restart the RootIO app when deemed necessary.

## 6.2 Downsides to using Android mobile phones

Even though Android phones proved to be an adequate prototyping platform in the communities I initially researched, attempting to use them for long-running broadcast proved challenging. Almost all the issues I encountered were down to the conception of the phone as an interactive, battery-powered device, designed with concerns regarding privacy, cost, and energy optimization. The RootIO app worked against these considerations in many ways, and necessitated operations that were more suited for a device designed for long-running daemon-like processes. Below, we look at the issues that hindered the use of mobile phones as a computing platform at RootIO transmission sites.

### 6.2.1 Resource optimizations

Optimizations to minimize battery drain by applications are refined with every successive Android version release. A notable pattern is that the operating system is constantly trying to send apps to sleep and to wake them once the user returns to the screens of those apps, or when trigger actions are performed, such as plugging in a charger, picking the phone up, or unlocking the screen. For long-running applications such as the RootIO application, this results in interruption and potential termination. To circumvent this, the user must explicitly go through the application settings and disable battery optimization for the RootIO applications.

Similarly, Android mobile phones embed the notion that mobile data is pricey and should be used sparingly by applications. The implication is that, past a certain threshold, the operating system denies particular apps access to Internet via mobile data to avoid extra charges. This is also the case when the phone determines that the Internet is being accessed via a roaming telephone network, irrespective of cases where the roaming is local and does not incur extra charges. As such, users had to explicitly disable Internet optimization for the RootIO app, especially in communities where streaming was used. The impact of this was that installation of the RootIO app was not a turnkey process, but rather a process that required advanced knowledge of mobile phone settings.

### 6.2.2 Application lifetime uncertainty

Unlike in a typical computing environment where long-running applications have more autonomy and only have to compete for the resources required by the operating system, applications on the mobile phone are more dependent on the underlying operating system and can be suspended and resumed depending on how much the user is interacting with them. In events where the Android operating system is starved of resources, the behavior is to suspend the application and resume it, resulting in uncertainty in the flow of processing, as opposed to the typical scenario in computing platforms that would merely result in delayed processing.

Given that the RootIO app was always playing audio content and managing phone calls, this resulted in frequent disruptions that manifested as one song stopping and another starting to play. In the case of events such as phone calls, a great deal of work was needed to resume the calls and restore the app correctly if it was suspended and resumed in the middle of ongoing phone calls. For the kind of routine use that is envisaged for mobile phone users, these issues are trivial because the user can always relaunch the application if it is terminated while they are using it. In the case of RootIO, however, there was need to monitor the communication from the RootIO app and trigger restarts of the app in cases where I noticed lack of communication in the stipulated time.

I also observed that the mobile phone performed different optimizations depending on whether the phone was charging or running on battery. The combination of energy, network, and computing resource optimizations made for a constantly fluctuating environment that does not favor the operations of the RootIO application.

### 6.2.3 Restriction of critical functions

Mobile phones are kept within close proximity of users and often bear private information that their users may not want revealed. The result of this is that mobile phone manufacturers are constantly improving the features geared towards ensuring user privacy, as well as preventing take-over by malicious applications. Some critical functions are thus kept out of reach of third-party applications. In the case of the RootIO app, one critical function that would have helped, as we will see later, is the ability to programmatically restart the mobile phone. It is a well-known fact that many issues that affect computing systems can be solved by regular restarts, which is the reason why common devices such as Wi-Fi routers and the GOIP devices I used for telephony all support scheduling routine restarts.

Another function that would have been beneficial, but is out of reach, is the ability to remotely access the screen of the phone to control it. In some scenarios RootIO stations were installed far from the community, in places such as hilltops, and this function helped in cases where it set up. In phones that supported unlocking the bootloader to enable these functions, it was possible to set them up, but the process was not simple and efforts to have the communities root the devices proved futile as this required technical skills that were far beyond those needed for ordinary use of a mobile phone.

Another observation is that with recent versions of Android many permissions are now granted only for times when the user is interacting with the phone. This adds another level of uncertainty when coupled with the previously mentioned resource optimizations. Again, this is not an issue for routine use of the mobile phone; the user can be prompted to grant permissions as is necessary, but for unattended use as required by RootIO, this was a major setback.

### 6.2.4 Lack of auto-deploy functionality

In the initial stages of development, I was able to host the RootIO app on Google's play store, but as previously mentioned, growing concerns meant that applications were subjected to stricter scrutiny and their functions evaluated for potential harm to mobile phone users. Given the extensive permissions needed by the RootIO app (SMS, calling, network information, Media libraries, External storage, SIP telephony, GPS location, recording), it was deemed a high-risk application and therefore banned from the play store. This was reasonable given the fact that the app was not intended for individual use and was published with clear warnings instructing users to not install the app on their individual devices.

The impossibility of having the application on the app store meant that there was no easy place to find the right packaged application to install it in the first place. This was evident in the various versions that were running in different communities as logged during synchronization sessions. The easiest solution to this was to create a shared Google Drive folder and create a shortcut to it on the different phones, and clearly include the version number in the application package name. In addition, the phone would announce its version of the app when communicating with the cloud server, enabling me to advise communities that were running older versions of the app.

This ban from the play store also resulted in the loss of the only way to automatically update the application. The Android operating system requires users to manually accept when installing or upgrading applications, unless done through the play store. This meant that changes could not be automatically pushed to communities, even though their transmission site had Internet access.

## 6.3 Switching to less restrictive devices: The Raspberry Pi

Initially, the use of computers with a full Linux kernel, or offering a full JVM, were investigated but not taken up, given the uncertainty around telephony integration and ease of use within communities. As the RootIO idea matured, however, and a clear use model emerged, the potential benefits of running on fuller computing devices with reduced energy demand and interconnection options became increasingly apparent. The obvious candidate was the Raspberry Pi [108], according to the evidence produced by my research. The Raspberry Pi was conceived as a low-energy board to add computing to routine operations, as opposed to PCs, which are designed for fuller stand-alone computing. The Raspberry Pi model is additive: one adds the components necessary to achieve the purpose for which computing is desired, which aligns well with my additive model of community radio.

Given that the Raspberry Pi runs a full JVM, the new approach is less dependent on the underlying operating system and hardware, but only bound by the possibilities of the JVM in use. Even though this setup provides some challenges of its own, these are much fewer when compared to the set of challenges posed by the mobile phone; and, more positively, there are ample solutions and design alternatives to mitigate the challenges in this latter setup. In the following sub-sections I discuss in detail the switch to the Raspberry Pi as a viable and promising alternative that eliminates the technical issues associated with the telephone. The Raspberry Pi also offers ample space to accommodate many of the implied needs from community members.

As has already been mentioned in chapter 5.6.2, communities had already incorporated Raspberry Pis and desktop computers into their broadcast ecosystem, meaning that targeting a full JVM, and in particular a Raspberry Pi, would not stretch the technological ecosystem in these communities. By this time, the advantages that came with using Android are clear to the reader, but I would like to highlight how I retained these advantages while moving to a fuller computing platform that adds possibilities to what was already offered by the Android platform. I also highlight how a consideration of these gains ultimately narrowed the available options down to Raspberry Pis, even though the options seem plentiful in this space.

### 6.3.1 Cost optimization

The Android mobile phones I used were mostly in the €50€ to €250 price range, much cheaper than a reasonable small laptop. This kept the price of the entire ensemble considerably low. Switching to devices that supported a full JVM and other computing capabilities signified a potential cost increase in this area. The Raspberry Pi version that I used came to a total of €120 including accessories such as casings, memory cards and power adapters.

It is worth noting that the more expensive mobile phones had more processing power, but this was because they also had other sophisticated features such as fingerprint readers and extra cameras, all of which drove up the cost of the phone but were not necessary for the purposes of RootIO stations. In the case of the Raspberry Pi, all these extras are available as add-ons and therefore have less bearing on the price of the Raspberry Pi unit itself, much to the advantage of the RootIO setup.

When it comes to the cost of operation, the main disadvantage introduced by computing devices is extra energy consumption. It is also the case that most computers are designed to work with AC energy setups and would need some adaptation, or extra equipment like solar inverters, to fit them in the ecosystem. Raspberry Pis, on the other hand, work extremely well in solar setups, given that they can be powered from a USB slot designed to charge a phone, even without the need for DC-DC step up.

In the majority of setups that I created, the solar charge controllers had dedicated USB-A ports for charging devices, but the RootIO setup was already designed to include USB charging transformers with adjustable voltage and currency settings. Given that the Raspberry Pi is a small device and has no screen or other battery-intensive features, it serves as a direct replacement for the mobile phone when it comes to energy consumption. The Raspberry Pi therefore does not stretch operational expenditure, but rather provides significant savings on capital expenditure

### 6.3.2 Telephony and networking concerns

Telephony was one of the things that Android mobile phones did very well, given that this is their primary function. My initial decision to not use the Raspberry Pi, or any other computing platform, was driven by the complexity that arises from integrating telephony with these platforms. The modem that I ended up using in order to utilize telephony has 256 AT commands, each of which allows the user to control a specific function that one would already find implemented in a mobile phone. Besides control concerns, work needs to be done to route the audio coming in from the telephony adapter.

However, not all the aforementioned AT commands are necessary to achieve the functionality that was required for RootIO's use scenarios. Only eighteen of the available AT commands are handled in the RootIO implementation and these relate to call management, SMS retrieval/sending and signal quality reporting. The communication to the modem is by means of a 9600 baud rate serial TTY connection via USB, and the audio from the telephony board is fed into the Raspberry Pi through a 3.5mm audio input jack. The full implementation is detailed later in chapter 6.4.1.

The telephony board also has provision for a GPS antenna, allowing for location information similar to what would be available on Android phones, and is also accessible via the same implementation that allows communication to the telephony modem. The board is also capable of providing Internet access, up to LTE speeds, and the Raspberry Pi automatically recognizes it as a network interface. While the board may not have Wi-Fi capabilities, the Raspberry Pi comes with inbuilt Wi-Fi and Ethernet network interfaces completing all the viable connectivity options. My research results in chapter 6.5 show that the telephony board is reliable and responsive over long periods of time.

### 6.3.3 Media options

Mobile phones are increasingly becoming entertainment devices, which means that they come with rich features for media play, including support for a wide array of audio formats, media cataloging, audio equalizers, and enhancements. What the Raspberry Pi lacks in ready-made media features, it makes up for by affording the user the freedom to implement and enhance custom audio pipelines. The Java Media Framework® allows for slotting media handlers for different audio formats into a uniform API. Moreover, using tools like SOX®, enhancements can be made on audio channels, allowing for additional processing on audio channels, especially when it comes to telephony channels on which content is previously optimized for voice content, and audio levels are generally lower.

The Raspberry Pi lacks automatic media cataloging, but this was very easy to implement, albeit in a rudimentary fashion. There are edge scenarios in which this implementation lags behind the Android alternative, but, as explained in chapter 6.5.6, in the long-term it serves the same purpose to build a media library similar to what is available in Android mobile phones.

### 6.3.4 Ubiquity

Raspberry Pis are relatively new on the scene and are mostly used in academic settings and by electronics enthusiasts. Even their deployment in distanced communities, such as in community networks as detailed by Baig et al [7], their use is by technologically savvy practitioners as opposed to individual community users. The resulting advantage is that they have undergone tremendous improvement and development in a short time, with very many HATs designed for all sorts of hardware integration. They have, however, received little use in distanced communities that would be the main beneficiaries of RootIO’s technology. As a result, the skills needed to operate them are expected to be in short supply within these communities, and this is a core assumption in the design of the RootIO stack for Raspberry Pi. Designs that use the Raspberry Pi therefore have to demand the least sophisticated interactions from users in distanced regions, such as restarting and re-inserting SD cards.

The Raspberry Pi offers full access to the underlying operating system, making it possible to automate functions that require members of the community to interact with the mobile phone. I utilize functions such as the capability to reboot the device and remote access via VNC or SSH, thereby lessening the need for manual intervention at the transmission site. Furthermore, given their price and their growing popularity within academic circles, Raspberry Pis are much easier to access in comparison to other devices such as FM transmitters, which were not in stock in the capital cities of the communities in which I worked.

### 6.3.5 Messaging and trigger options

The Android "intents" messaging system detailed in chapter 6.1.5 allowed for smooth inter-process communication. In the Raspberry Pi version this would have been possible with the use of an AMQP [113] compliant messaging service such as RabbitMQ, but this was deemed very demanding for the specifications of the Raspberry Pi. Instead, the RootIO application on the Raspberry Pi is a monolith, with each module launched in a separate thread but accessing a shared register that contains references to all the other modules and can thereby invoke the message processing method of another module to trigger a change in state. To date, this has worked well, linking the SIP and PSTN telephony modules with media play modules. All the asynchronous messaging that was used in the Android app has been replaced by synchronous method calls using shared references.

Most of the events that were relied on to perform some recovery app actions do not exist in the Raspberry Pi. The option to restart services and reboot the device through SMS is still available through the use of the telephony modem added to the setup. I also implemented the app as a Linux service with recovery using systemd [14], which means that in the event of the app crashing, it is restarted after a set amount of time. A last-resort effort is an automated nightly reboot of the Raspberry Pi courtesy of the Cron service [85], resetting any buffers and counters that might pile up over time and result in degraded performance. The results I obtained show that this is particularly useful in the case of SIP telephony, given that SIP-related program failures have greatly declined after this change was implemented, and that the SIP module is always registered with the telephony server in the RootIO cloud platform.

### 6.3.6 Improvement on resource optimization failures

Android phones are actively looking for ways to minimize battery consumption, whereas the Raspberry Pi does not have a battery in the first place, but also aspires to perform at the level expected of a server platform. Moreover, the Raspberry Pi lacks any of the optimizations that occur due to low battery, change in the network status, or accumulated network use. The separation of telephony into a separate board greatly reduces the processing needs of the Raspberry Pi and leaves more resources available for use by the RootIO application.

The Raspberry Pi, unlike the mobile phone, does not suspend and resume the RootIO application, but rather terminates it in extreme circumstances, or delays the processing in cases where there are many processes vying for resources. The considerable advantage of this setup is that the flow of processing is deterministic. Crashes and delays have, however, not yet been observed because the Raspberry Pi comes with very little software pre-installed and therefore there are almost no applications contesting for resources, as opposed to the myriad of apps that are usually pre-installed on smartphones, some of which cannot be disabled by the user. In the event of crashes, however, systemd will resume the RootIO application.

### 6.3.7 Improved deployment and remote monitoring options

The Raspberry Pi affords numerous ways to deploy and update the RootIO app. For a first practical approach, I created an installable version of the Raspbian® operating system that comes bundled with the RootIO app. All that users have to do is burn this image onto a memory card, boot the Raspberry Pi, and edit the configuration file to associate the Raspberry Pi with a particular FM radio station hosted on a defined RootIO cloud.

Communities may lack the facilities and knowledge to manipulate the Raspberry Pi device and so another viable option is to make use of an option in the RootIO platform that allows them to generate and download a Raspbian image pre-configured for a particular radio station. In this scenario, the configuration that maps the Raspberry Pi to a station, as well as the details regarding how to connect to Wi-Fi networks would be baked into the image. The Raspberry Pi can then boot from this image, connect to the specified Wi-Fi network, and thus start communicating with the RootIO cloud.

A third option that works well for bulk installations, but is mostly used by people with technical knowledge, is to use Ansible [46], an Infrastructure-as-Code tool. In this scenario, once an operating system is installed on the Raspberry Pis and they are connected to an IP network, a user can run a script that configures all the specified devices and starts the RootIO app on all of them. This was my preferred approach given that it allows for flexibility to perform extra actions, as well as fetch extra information while doing the installation. Another advantage is that the same base image is downloaded once and customized for all the other installations, thereby saving bandwidth, as opposed to the second option which requires a different image for each station. Eventually the scenario used will depend on the circumstances in individual communities.

When it comes to updates, the Raspberry Pi can simply download a newer version of the RootIO app from the online repository and install it in place of the old one, without requiring any human intervention. This means that within a short time of uploading a new build of the RootIO app, all the Raspberry Pis with Internet access can have the latest version of the application running. Currently, the update script only has to check for a new JAR file in a specific online repository, but this can be made more robust to rely on Debian’s package management utility.

Regarding remote control and monitoring options, the Raspberry Pi has many more usable options, given that it is in many ways a full Linux computer. In areas where Internet is available, the Raspberry Pi creates a reverse SSH session to the RootIO cloud platform. This in essence makes it possible to access the shell on individual Raspberry Pi devices for purposes of troubleshooting and performing restorative actions, which can also be automated. Where a GUI is needed, the Raspberry Pi comes with a VNC server that allows for remote access to its display, if there is an Internet connection. One possibility is for the RootIO cloud to provide centralized VNC access to all the connected devices, allowing even the initial configuration to be achieved in the RootIO platform.

## 6.4 Architecture of the RootIO setup on the Raspberry Pi

As previously mentioned, the decision to run the RootIO app on the Raspberry Pi was made in a bid to retain the advantages afforded by the Android mobile phone. Cost, energy, and portability advantages are all shared between the Android mobile phone and the Raspberry Pi, but there are demonstrable performance and stability gains running on the latter. The reader should remember that the RootIO application is capable of running on any device with a full JVM supporting Java version 8 upwards, is capable of serial communication over USB-A ports, with a line-in interface for receiving audio input, and capable of hosting a MySQL database.

I developed and tested the improved RootIO application on a Windows 10 computer and the transition to the Raspberry Pi was seamless and needed only changes in a configuration file, given the portability afforded by running on a JVM. The architecture that I detail below is therefore possible on any device with the aforementioned capabilities, though in my thesis I specify the Raspberry Pi because of the affordances outlined above.

### 6.4.1 Telephony integration

Integrating telephony into the Raspberry Pi was always going to be a challenge given that telephony is an intrinsically sophisticated concept. My initial attempts at what would have been a straightforward integration yielded poor audio quality, even while using hardware that had worked well for other projects where I used more powerful computers. The alternatives required a tremendous amount of work at a time when I was not certain it would end up in the final product which would be used by communities. Below I detail three integration approaches that I studied, and in which I have gained expertise over the course of my research. I examine the advantages and drawbacks of each, arriving at an optimal setup that has delivered promising results.

#### 6.4.1.1 Telephony by use of a USB dongle

My initial attempt at integrating telephony involved the use of a USB dongle. These are commonplace in developing regions where fixed telecommunication networks are scant and telephony is almost entirely mobile. These allow Internet access mostly to personal computers using the same network technologies relied on by mobile phones for Internet access. The obvious advantages were that these are ubiquitous, portable, and require no energy considerations beyond what can be supplied by a USB-A port.

The USB dongle once inserted into a computer, will create a serial port for signaling and another USB port for audio data. While working on a project for the Deutsche Welle Akademie in Uganda that used these modems, I noticed that there was often a reset when the modem switched network technologies from 3G to 2G technologies. This was usually due to the dongle demanding more current amperage than was available on a given USB port. The result of this was that there was no guarantee regarding the paths at which the audio and signaling channels were mounted, yet this had to be specified in a configuration file to the system controlling telephony through this modem. A fix for this problem was to detect the reset, rewrite the telephony configuration and restart the telephony system, coupled with using a powered hub to connect the dongles to the computer.

Another drawback of this setup is that not all USB dongles have support for voice telephony. While all of them have supported access to mobile Internet, from my previously mentioned work with Deutsche Welle Akademie, I discovered that only a few models, mostly from Huawei, were capable of delivering voice telephony. As can be seen at this link: http://freeswitch-users.2379917.n2.nabble.com/mod-gsmopen-now-with-USB-dongle-support-SMSs-and-Voice-calls-to-from-Cellular-Networks-td7484987.html, users had to constantly share tips on which dongles, by which manufacturers, were capable of delivering voice telephony.

The last drawback is that in almost all cases these modems were manufactured for particular telecommunication companies and therefore required a special procedure to unlock them to work with the SIM cards of other preferred providers. This meant that the chances of success using this alternative were contingent on finding the right dongle and guaranteeing its continuous operation, a big ask for the distanced communities in which I was conducting research.

In order to use these modems, one has two options: the first is to use a soft PBX system such as Freeswitch that has support for them as a telephony channel through an add-on module. This offers seamless telephony integration and Freeswitch is capable of routing telephony events to a local application, as well as routing audio coming from the modem. In this scenario, no code is needed to communicate with the modem, besides the standard call control commands utilized by Freeswitch. A soft PBX, however, has considerable demand on computing resources and is less than ideal for a micro-computing platform such as the Raspberry Pi.

USB dongles are cheap, with the last dongles I purchased costing little more than 10€. Furthermore, among all the alternatives they present the least hardware to work with and can add mobile Internet to the entire setup. Despite all these advantages, it was decided to not use them for the following reasons: they necessitate a powered hub if they are to function reliably, they do not perform any media processing of their own, but rather depend on the Raspberry Pi, and require extra software for telephony. These were the key factors that influenced my decision to not use USB dongles.

#### 6.4.1.2 GOIP (GSM to VOIP converters)

Another option considered was the use of GOIP devices, which I was already using for adding telephony to the RootIO cloud in scenarios where telecommunication companies were not willing to offer direct SIP interconnection with their switching infrastructure. As elaborated in chapter 3.3.1, GOIP devices allow a user to slot in a number of SIM cards that allow for telephony via PSTN networks and convert this to SIP telephony that is then relayed to a soft PBX via an IP network.

In the setting that I was using, the GOIP device would have been plugged into the Raspberry Pi with an ethernet cable, and would have been configured to communicate with a soft PBX (FreeSwitch) installed on the Raspberry Pi. The communication to the RootIO application past this point would be exactly the same as in the above-mentioned scenario with the USB Dongle.

The GOIP device has a wealth of valuable configuration options, and above all affords the possibility of using multiple SIM cards, therefore allowing the transmission device to receive more than one phone call. There are scenarios in which this could be useful, such as local conferencing scenarios, which do not go through the cloud setup but rather allow a facilitator to locally initiate a talk show program via conference calling.

However, I did not consider the GOIP option because this adds a sophisticated piece of hardware whose configuration and administration are not inconsequential. Additionally, GOIP devices are only available online and cannot be sold to some markets because of their association with SIM box fraud [4]. There is also the need to install a soft PBX on the Raspberry Pi, which I was already trying to avoid given the processing overhead. Taking into consideration energy constraints, GOIP devices require significant energy compared to the other telephony options: at the most basic level they consist of a Linux kernel that is doing both PSTN and SIP telephony, a power-intensive combination.

A last but more severely damning revelation from the use of GOIP devices in Romania was that they support only 2G networks, which are being decommissioned and were not offered by some providers on whom we relied. For these reasons it was decided to not use the GOIP. It is worth noting though that Deutsche Welle Akademie solved their USB dongle-related issues by switching to GOIP devices, a solution facilitated by their reliance on grid energy and use of full personal computers.

#### 6.4.1.3 Telephony HATs for Raspberry Pi

The Raspberry Pi, as mentioned previously, is a favorite of electronics enthusiasts, with the concomitant advantage of a wealth of HAT solutions for almost any perceivable computing need. Options exist for telephony, supporting up to LTE networking technology, capable of supplying voice telephony, Internet and GPS information. The existing solutions can connect via the Raspberry Pi’s GPIO pins, USB, or both. I chose the SIM 7600X 4G HAT which has the option to connect via USB because this required the least coding implementation to integrate with the RootIO app.

The SIM 7600X 4G HAT is capable of furnishing connectivity on 2G, 3G and 4G networks, providing a solution that bests the GOIPs, which were entirely restricted to 2G networks. It connects to the Raspberry Pi by use of a micro-USB cable which also delivers power to the device. Regarding audio, the user has the option to either retrieve a PCMU encoded stream via USB or use a male-male 3.5mm audio cable to connect the audio output of the board to an audio input on the Raspberry Pi, thereby allowing treatment of the board as an audio input device.

The board also has the capability to receive a GPS signal by plugging in an external antenna, allowing for access to granular location information that can be used to report the location of the RootIO station setup as a prerequisite for precautionary actions, such as shutting down the station once it is determined to be in a location far from that approved by the licensing regulations.

None of the other options for telephony integration offer the versatility of the SIM 7600X 4G HAT given its connectivity options, audio handling and additional location information. The one drawback of the SIM 7600X 4G HAT is that it requires the most coding work to integrate with the RootIO platform, a situation complicated by the fact that communication between the two is via a serial interface. The SIM 7600X 4G HAT once connected and booted is accessible on several serial ports – one dedicated for AT commands and the other for streaming PCMU audio. In my case I only needed the control port for AT commands, given that I was accessing call audio from the physical audio jack interface.

Communication with the SIM 7600X 4G HAT is asynchronous – the RootIO application configures it to start streaming information about call status, incoming SMS, and GPS location. The RootIO application then listens for messages and carries out necessary actions in the case of SMS messages and phone calls, or simply logs the information in the case of GPS location updates. If the RootIO application intends to terminate a phone call or send an SMS, it will initiate this action and listen for the respective response before executing the specified command.

Incoming SMS messages are handled in exactly the same way as in the Android mobile phone setup: the message is parsed to see if it contains any commands to manipulate the RootIO app, such as starting or stopping particular services or rebooting the device. These messages are then stored and synchronized to the cloud platform. Given that I am not using the PCMU audio stream from the SIM 7600X 4G HAT, it was not possible to include audio play of phone calls inside the RootIO application as was the case for the Android mobile phone, instead this is handled outside the application.

An incoming call prompts the sending of an event to the RootIO application, which in turn results in the pausing of any music that is playing. The RootIO application then instructs the SIM 7600X 4G HAT to answer the phone call, upon which audio is played into the Raspberry Pi, as mentioned previously. Using SOX, audio is routed from the audio input of the Raspberry Pi to its audio output that feeds the FM transmitter, thereby relaying the call to the community listening in.

```
private void playCall() {
    try {
        Runtime.getRuntime().exec(String.format("%s -r 8000 -c 1 -t %s %s -b 16 -t %s %s",
                Configuration.getProperty("sox_path", "/usr/bin/sox"),
                Configuration.getProperty("audio_driver", "alsa"), Configuration.getProperty("audio_input_device"),
                Configuration.getProperty("audio_driver", "alsa"), Configuration.getProperty("audio_output_device", "-d")));
    } catch (IOException e) {
        Logger.getLogger("RootIO").log(Level.WARNING, e.getMessage() == null ? "Null pointer[PhoneService.playCall]" : e.getMessage());
        //maybe hangup?
    }
```

**Figure 6.1: A code snippet for routing call audio using the SOX tool**

This approach relies on the inbuilt ALSA driver and as illustrated in Figure 6.1 would result in a command such as `"/usr/bin/sox -r 8000 -c 1 -t alsa hw:1.0 -b 16 -t alsa hw:0.0"`. This instructs SOX to play the input from audio device identified by id hw1.0 into device with id hw:0.0 at a rate of 8000 bits per second (-r 8000), with one channel (-c 1), using the ALSA driver (-t alsa) with a sample size of 16 bits (-b 16). The SOX process is started in the background and when an event is received signaling the end of the call, this routing is disrupted by halting the background SOX process. The one advantage of using SOX is that it is possible to add arbitrary effects to the processing pipeline to enhance the sound quality of the call.

## 6.4.2 SIP telephony integration

The reader will remember that while the initial RootIO concept was developed for communities with limited access to the Internet, it was also applicable in rural communities in Romania where there was sufficient Internet bandwidth. This resulted in the incorporation of some Internet-based approaches into RootIO's core design. One of these was the use of IP telephony using the SIP protocol. Given that RootIO's cloud already uses SIP for interconnection with PSTN networks that eventually link it to the individual transmission points, it was possible for the transmission point to connect directly with the cloud server via SIP.

In this setup, the Raspberry Pi is treated by the soft PBX in the RootIO cloud as an extension that can receive calls. The benefit of using SIP telephony as a channel for streaming audio content is that there are no equipment or telephony providers in the middle that may cause problems. All outgoing telephony from the soft PBX is via SIP telephony, and it is down to the implemented dial plan to determine whether to reach the phone through a direct SIP leg, or via a SIP leg bridged to a PSTN leg and eventually to the SIM 7600X 4G HAT if there is no Internet connection to the transmission site. Either way, the logic is the same, while if one were to use a streaming technology such as Icecast, different logic would have to be implemented to fall back to telephony in absence of Internet.

Adding SIP telephony to the Raspberry Pi did not prove to be easy. Given the complexity involved, I believed that the best solution would have been to depend on a well-developed SIP phone, such as Zoiper® or Linphone®. None of these, however, had the option to send events on call events, and would therefore have been impossible to integrate into the flow of the RootIO application. Eventually, I came across an application, Twinkle (http://twinkle.dolezel.info), that had this functionality, and it was incorporated into the RootIO flow, but this did not work well and eventually I implemented a rudimentary but stable stack in the RootIO application.

#### 6.4.2.1 SIP integration using a 3rd party application

I came across Twinkle while looking for softphones that had the capability to trigger a script if an event occurred relating to change of state of a call. This allowed for the RootIO application to be notified when a call was coming in or terminated, triggering, respectively, the halting or resumption function of any audio content that the RootIO application was playing, either from local content or an Icecast stream. The Twinkle application had the added advantage that it could be configured to automatically answer all incoming phone calls, as long as the script triggered did not direct the call to be declined.

The resulting architecture saw the Twinkle application invoke a Python script with call information every time a call rang, was answered, or terminated by either the caller or the callee. The Python script, in turn, communicated via a TCP socket to the RootIO application, which would return an appropriate answer depending on the caller. The use of an application such as Twinkle in this kind of architecture has the advantage of potentially adding telephony capability to a program written in any language, without having to worry about the available telephony libraries for that particular language.

When an incoming call was received, the Python script would connect to the RootIO application, which would in turn check if the calling number was present in the list of numbers permitted, and, if so, broadcast to any active media players an instruction to pause the currently playing content. Subsequently, the RootIO app would send back a JSON response with the "status" value set to "ok". The Python script needed to simply print `"action=autoanswer"` to the standard output and exit, for Twinkle to accept the call. In the event that the calling number was not allowed, or that the call was from a server other than that to which Twinkle was registered, the RootIO application would reject the call by setting the "status" value in the JSON response to "fail", to which the Python script would react by printing `"action=reject"` to the standard output for Twinkle to reject the call.

One drawback of the Twinkle application is that it runs with a GUI and cannot be launched on a headless system, or via a terminal without a display. It does ship with an interactive version designed to be run on a console, but this does not support running in the background, as the application exits as soon as the user's session is closed. It was therefore not straightforward to work out how to daemonize the application, nor how to start up when the Raspberry Pi was rebooted. A solution was to use the "screen" command that creates a session that persists even when the user closes their current session. This was achieved with the command:

```
screen -S twkl -X stuff 'twinkle -c -f /etc/twinkle/raspi.cfg \n'
```

The above command creates a screen called "twkl" (-S twkl) and tells it to execute the command in quotation marks when launched. This was then wrapped in a systemd service and could be launched when the Raspberry Pi started up.

During the course of running with Twinkle as the SIP phone of choice, I noticed many crashes of the app that could not be explained by the logs, as well as several instances in which the Twinkle application crashed and could not register with the soft PBX in the cloud. I therefore came to the conclusion that even though this was a working approach, it did not guarantee the kind of stability to match what was available on Android phones using the LinSip library. Even the introduction of nightly restarts of the Raspberry Pi were not enough to guarantee the service of the Twinkle application until the next reboot. It was for this reason that I abandoned Twinkle as a viable alternative.

#### 6.4.2.2 Implementation of SIP using the Peers library

There is a plethora of SIP libraries, but most of them would have required a lot of work to integrate into the RootIO application on the Raspberry Pi. To begin with, some of these libraries are written in a different language, making their incorporation into the RootIO codebase tedious. Other libraries such as PJSIP and Jain SIP only have support for SIP signaling, but no media capabilities, given that they do not have implementation for the RTP and RTCP protocols used to deliver the audio for SIP calls.

I chose the Peers library (https://github.com/ymartineau/peers) because it is a pure Java implementation, guaranteed to work in different environments; as long as a JVM is installed, it can run. This saw the SIP functionality move into the RootIO application, similar to the setup with the Android version of the app. This allowed for finer control and monitoring of SIP calls within the RootIO application.

On startup of the RootIO application, the SIP client attempts to register with the soft PBX in the RootIO cloud platform. Once successfully registered, the RootIO application in the Raspberry Pi is effectively an extension to the soft PBX, as would be the case with the Twinkle application. The cloud platform can then call the Raspberry Pi directly via SIP, and specific events are triggered in the RootIO application, depending on the triggering call events and state. As previously mentioned, messaging within the RootIO application is through shared memory, in the form of callback functions on particular objects.

On receipt of an incoming call event, the SIP telephony module will check that the call is originating from the server to which the SIP client is registered. If this check passes, the media module of the RootIO application responsible for playing audio content is notified to stop or pause any audio content that is currently playing. The media module will typically fade the music out over a predefined period, during which period the answering of the call is delayed by the SIP client.

Upon answering the phone call, an event is broadcast by the SIP client notifying that there is now an ongoing call. This sets a global status that is checked in edge cases, such as when, in the background, another local program is scheduled to start but should wait to play audio because there is still an ongoing call from the cloud server. The reader should be reminded that the cloud platform and the RootIO application running on the Raspberry Pi each run their individual schedules, but the RootIO application yields its programing every time a call comes in from the cloud platform.

When the call from the cloud platform is terminated – normally because the cloud platform has played all the content that there is to play – the SIP client broadcasts an event signaling the end of the call. This prompts the media module of the RootIO application to resume playing the content that was paused when the call came in, or to launch new content in the event that the program in the background changed during the call.

The implementation of the SIP client in the RootIO application is rudimentary at the time of writing this thesis but has worked very well. Challenges remain, however, such as the inability to tell when the local IP address changes due to a change of network adapter, or even the assignment of a new address on the same adapter. This address is used in the registration process, which fails if said address is not the one currently in use. This is easily fixed by first re-running the IP address discovery, in the event that registration fails, and registering much more frequently, at the expense of using more bandwidth.

Another drawback is that in its current state the library only has support for PCM-U and PCM-A codecs, with no support for any of the higher quality codecs usable for SIP calls, such as Opus. This in turn diminishes potential gains from using IP networks and can be detrimental to the listener experience if used extensively to relay programs that contain rich media, such as music. The user should be reminded that PCMU and PCMA are optimized for human speech and are not even used in modern LTE networks that depend on the richer EVS codec [29]. On a more positive note, the implementation has shown proper traversal of NAT networks, an Achilles' heel of SIP telephony. The implementation also showed good recovery performance in a simulation of a prolonged loss of connectivity to the soft PBX spanning multiple days.

## 6.4.3 Media play and management

Managing media on the Raspberry Pi was one of the more difficult implementations. It was necessary to both keep an up-to-date catalog of media, as well as support various media formats, since my analysis of music files gathered from communities in Northern Uganda showed that five file formats (.wav, .mp3, .aac, .wma and .flac) were in common circulation. Below I detail measures taken to both implement playback for these formats and maintain an updated media library.

### 6.4.3.1 Maintaining a media library

An aspect that I foresaw being a challenge, but that turned out well, was the implementation of a versatile media playback function. This is something that modern smartphones do well, given that they are designed largely as entertainment devices. On the Raspberry Pi, this posed a three-part problem: Keeping track of a media library and updating it when new media files are added, building a media player with the capability to play the vast amount of audio files in various formats in circulation, and the ability to play audio streams from online locations.

Building and maintaining an updated media library was a challenge because while on Android this was performed automatically on certain events, there was no easy way to recreate these events on the Raspberry Pi. The procedure to build the media library is detailed in chapter 6.5.6, but it is a basic recursive traversal of directories defined to contain media, adding discovered media files to a table in a MySQL database. I propose three plausible alternatives, which are not mutually exclusive; they can be deployed together to best recreate the media cataloging functionality present in Android.

The implemented option at the time of writing this thesis is a rudimentary approach in which the media library is rebuilt after a configurable amount of time. To achieve this, the media library module starts by obtaining the last date at which media was indexed, by querying the MySQL database in which media information is stored. If the last date is before a defined time window, or does not exist, a new synchronization procedure is started. This creates a new media library in a temporary table, clears the data from the existing media library table, and copies the new catalog over from the temporary table.

Another possibility is to harness the possibilities provided by udev [15] rules. The Linux subsystem provides udev rules as a convenient way to trigger actions when a device matching certain criteria is connected to a computer. The reasoning here is that media is most likely to be transferred using a USB stick, and therefore the ejection of a USB stick results in the possibility that the user just transferred audio files onto the Raspberry Pi. Given that the Raspberry Pi does not have a screen attached, udev rules can be used a step further to automatically copy media files into the library once a USB stick is inserted, and the library rebuilt once the USB stick is ejected.

Monitoring USB device inserts and ejections, however, would not be sufficient to detect files transferred to the Raspberry Pi via IP networks, such as by file downloads or SCP and FTP transfers. A solution for this is to monitor the size of the directories that are scanned when building the media library and initiating new media library indexing when the size changes. In this scenario care must be taken to not include directories that constantly change, such as those used for logging and temporary files, in the list of paths monitored by the media library module. The “Watch Service API” in the java.nio.file package provides an easy solution to incorporate this into the RootIO application.

Combining the three above-mentioned approaches recreates media indexing at a par with what is currently offered by the media library function in Android, but also provides a very efficient way to transfer files to the Raspberry Pi. The first time that the application is installed, the media indexing process is performed. Whether or not automatic copying of audio files from USB drives is enabled, if a file is copied into directories defined as holding media for the RootIO app, reindexing is performed.

#### 6.4.3.2 Media play

For media playback I had to implement a media player, given that the media player functionality bundled with JSE gives basic functionality to play only wave audio format files, yet from experience most of the audio files circulating within communities were in MP3 and other formats geared at achieving better quality/size tradeoff. As the reader will recall, a similar challenge on the Android platform was addressed by use of the Exoplayer library. To incorporate media playback into the RootIO application, the best approach was to rely on the JMF framework.

The JMF framework specifies a common set of functions, and in turn, relies on specific libraries to utilize those functions for different audio formats. It is easier to perceive JMF as a chassis into which a driver is slotted for any desired individual media format. The audio drivers I added facilitated the RootIO application to play MP3, WAV, OGG, and AAC audio formats which, in my experience, were the most common formats of downloaded audio files.

Though playback for Icecast streams was implemented, it was not as complete as what is available on the Android platform. Streams playing in MP3 format are choppy, a defect that was observed on other platforms and thereby ruled out hardware insufficiencies on the Raspberry Pi. However, streams in WAV format work perfectly, but again MP3 streams are popular for the same reasons as MP3 files: they use far less bandwidth to deliver the same perceived quality to the human ear.

### 6.4.4 Diagnostics and synchronization

The Raspberry Pi yields cleaner diagnostics, given the various Linux APIs for accessing granular processor, memory, network, and storage information. It was thus easier to get a clearer picture of the actual processing demands of the RootIO application and supporting software. It is possible to call or execute shell commands or scripts, respectively, which give detailed and granular information about vital metrics, such as a list of all running processes and network connections, as opposed to the Android platform where these were not accessible through the APIs provided through the SDK.

The diagnostics module of the RootIO app, combined with the synchronization module, work together to keep the cloud platform – and therefore the administrators of RootIO stations – up to date on the status of the different Raspberry Pi devices. On the Android mobile phone, the synchronization module was always hindered by the phone’s limits on Internet use by specific applications, especially when certain thresholds were crossed. The phone also always tried to send this module to sleep because it ran in the background. On the Raspberry Pi, both these restrictions do not exist, and therefore the synchronization module was much more reliable and is usable as a heartbeat monitor for the associated device.

## 6.5 Performance of the RootIO modules on the Raspberry Pi

In chapter 3.5, I showed that the RootIO application was initially divided into five services or modules that perform the different functions necessary to deliver FM broadcast at the transmission site. The reader will also remember that some of these functions were merged or transformed into entry points launched by the Android operating system on the mobile phone.

On the Raspberry Pi, these functions are organized differently, but given the fact that I am using a more reliable inter-process communication implementation based on shared memory, the split is reintroduced between all services. The emergent setup is as below:

- Telephony Service: This is responsible for PSTN telephony via the SIM 7600X 4G HAT. On top of monitoring and managing call status, this module also handles communication with the SIM 7600X 4G HAT over a serial port. This connection also facilitates SMS messaging and retrieval of GPS information. Given its added functions, this is detached from the radio service in the Raspberry Pi.

- Radio Service: This module observes the defined program schedule, playing the specified content at predefined times, either from local files or online streams.

- SIP Service: This module is responsible for SIP telephony. In the Raspberry Pi, this is implemented using the Peers SIP library.

- Diagnostics Service: This module is used to obtain metrics about the network, memory, processor, and storage resources of the Raspberry Pi.

- Synchronization Service: This service is responsible for synchronizing the data collected by the diagnostics service, call and SMS information, and the media library to the cloud platform. In return, this downloads whitelists of numbers allowed to call the transmission site, program schedules and playlists. This service also downloads configuration information for how to connect to the cloud, as well as how often to take diagnostics, how often to synchronize with the cloud server, and SIP configuration information.

- Media Indexing Service: This service periodically scans predefined directories for any newly added media and updates the media library. This service was introduced specifically for the version of the app running on full JVMs.

Ultimately, the suitability of the Raspberry Pi as a transmission site device comes down to how well it runs the six services that make up the RootIO functions at the transmission site. These were observed over a period of nine months for various performance metrics, and below I present the results that demonstrate that the Raspberry Pi solves the technical issues that plagued the RootIO setup when Android mobile phones were used.

### 6.5.1 Setup of the study to observe the performance of the Raspberry Pi platform

To analyze the performance of the RootIO system on Raspberry Pi devices, I set up four identical devices in the same environment, connected to a Wi-Fi network that facilitated their communication with the cloud platform. The devices were observed over four distinct observation periods, each lasting at least a month. The first window was from 9th September 2020 - 24th October 2020, the second observation window was from 26th October 2020-28th December 2020, the third observation window was from 1st January 2021-14th February 2021, and the last observation window was from 16th February 2021-31st May 2021.

In between observation windows, changes were made, and their effect observed in the next window. One of the devices suffered an SD card failure early in the first observation window and was therefore not contributing at all to the statistics in the first observation window. It was, however, possible to obtain consistent data from the remaining three devices. One of the devices also had special hardware, which we expect as part of the final assembly such as RTC clock modules (to store time information across reboots) and the SIM 7600X 4G HAT. The former did not have any impact on the performance of the device to which it was attached, while the latter facilitated an observation of the performance of the telephony service on one of the Raspberry Pis.

I was observing the performance of the six services that are core to the operations of the RootIO stack at the transmission site. As a whole, the system was monitored for its capabilities to play media, handle phone calls, and communicate with the cloud server when conditions were ideal. I also tested recovery scenarios simulating long Internet outages, outages of the web server, and outage of the soft PBX. In all these scenarios the Raspberry Pi demonstrated stellar performance in ideal conditions as well as the ability to recover when conditions deteriorated but eventually resolved. Below I elaborate on the performance of the six RootIO modules and show why the Raspberry Pi is a better computing platform than the Android mobile phone at the transmission site.

### 6.5.2 Performance of the Radio Service

The radio service is critical because it manages the schedule and plays audio content. In the entire RootIO ecosystem this module contributes the most to user experience given that, from my findings, transmission-side programs accounted for more than 80% of broadcast time, even in communities in Romania that were heavily dependent on cloud-based programs. The performance of the radio service on the Raspberry Pis was assessed from 630,750 datapoints created by events relating to media play.

In observing the performance of the radio service in the fourth observation windows, consistent performance was observed by analyzing the frequency and distribution of the START, STOP and PAUSE events created by the radio service. The START event is created every time that the radio service starts to play a song from a local media file or an online Icecast stream. The STOP event is created when a media file has played to the end or is ordered to stop playing because the program in which it was playing has come to an end. The PAUSE event is created when a song or stream is interrupted due to an incoming call either via SIP or PSTN telephony.

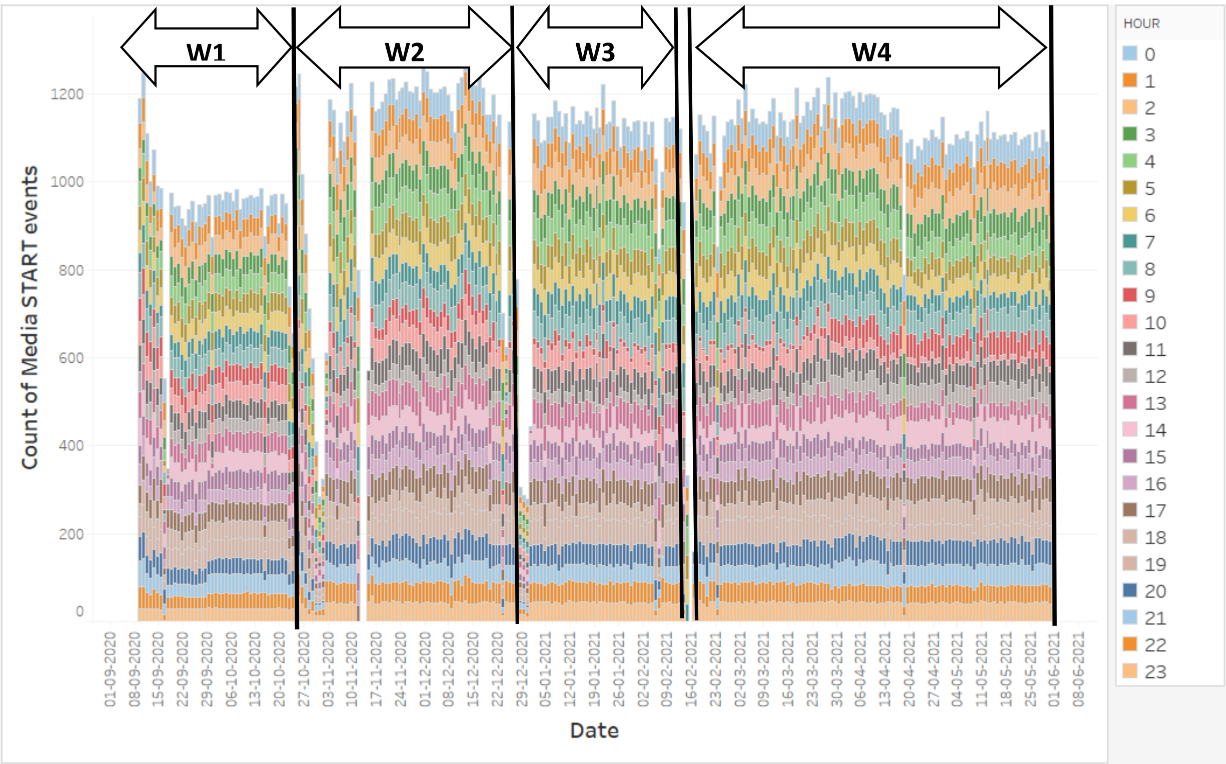



**Figure 6.2: Figure showing number of media START events across the 4 Raspberry devices during the 4 observation windows**

Excluding the device that suffered a disk failure, it was observed that on average the other three devices showed an almost uniform distribution of these events across different days, averaging 1061 START, 1287 STOP and 26 PAUSE events daily across the f observation windows combined. The minimal variations on individual days were due to differing lengths of songs, and the fact that each day has a unique combination of songs, coupled with the fact that cloud programs last different amounts of time due to the varying duration of the content they play. The schedules are also different on different days in some stations.

It is worth noting that while the START to STOP ratio should ideally be 1, for legacy reasons relating to flaws of the process on Android, it was deemed necessary to stop the media even after pausing it due to an incoming call. This was because the paused media player would get recycled by the operating system and could not be resumed as would be expected. A prudent approach was to go on and stop the media player and discard it behind the scenes, and log a PAUSE event to advise that the associated

STOP event was due to interruption by a call. Moreover, in the event that a song is terminated because the program in which it was launched has come to an end, a STOP event is generated, and the media player also generates a STOP event upon exiting. In another scenario, because cloud programs are scheduled at the top of the hour, the cloud may call a few seconds into the start of a new program playing locally on the Raspberry Pi. This program, which would have only barely terminated the previous program, will also need to pause its media, resulting in a quick succession of extra STOP events.

The START event is therefore a credible proxy for the analysis of the performance of the Radio service as its distribution is symbolic of when and how the radio service is running. For the Raspberry Pi setups, an average 1061 START events were observed daily with a median value of 1124 and a standard deviation of 196 over the four observation windows. In the last, and most steady window, the mean, median and standard deviation were 1106, 1117 and 135 respectively, suggesting increased stability of the radio service. The noticeable drop in events in the week starting on the 20th of April 2021 was due to the introduction of longer content in cloud programs, resulting in less airtime for music programs.The distribution of START events by time of day, proves that the station is functioning round the clock. There is also a consistent distribution between local media and online streams, showing that the radio service performed well.

## 6.5.3 Performance of the SIP service

The SIP service was one of the harder modules to implement on the Raspberry Pi, but the results obtained vindicate the choices made. When I relied on Twinkle as a softphone, it demonstrated consistent performance when it was running, but was prone to frequent failures. These resulted in the introduction of nightly restarts but, even then, there was no guarantee that the service could last until the

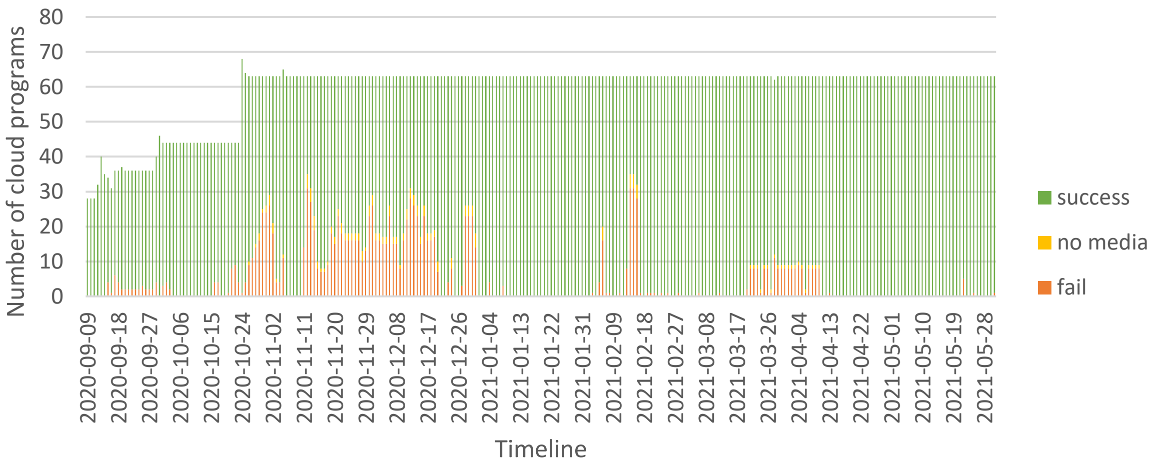



**Figure 6.3: Graphic showing performance of cloud programs across the 4 observation windows**

next restart. Worse still was the fact that no reason could be found in the logs for the Twinkle application ceasing to run. The rudimentary implementation I made for the Raspberry Pi is far from what is expected of a softphone but bears the basic functions to support the RootIO use case, and these performed well.

On switching to a custom implementation via the Peers' library, an analysis of the SIP service from 1,835,782 data points collected during two observation windows demonstrated much better performance on programs originating from the cloud as well as consistency in registrations with the cloud server. As can be seen from Figure 6.4, there was very consistent performance in the last observation window (w2) with one noticeable dip due to daylight savings time change (timestamps shifted by 1 hour), on the 25th March 2021. The other noticeable outage on the 8th of May 2021 is characterized by records missing for entire hours, synonymous with a general network outage affecting all stations.

The Raspberry Pi also benefitted from the lack of power and network optimizations, as was the case with Android, resulting in much more stable and uniform performance across multiple observation windows. The SIP service implementation also demonstrated resilience in the event that the Raspberry Pi moved into non-ideal conditions for a long time. These included loss of Internet for prolonged periods, sudden reboots, and the loss of connectivity to the soft PBX in the cloud. The period between observation windows 2 and 3 was used to simulate a prolonged Internet outage. It was shown that communication with the cloud server was restored as soon as Internet was available.

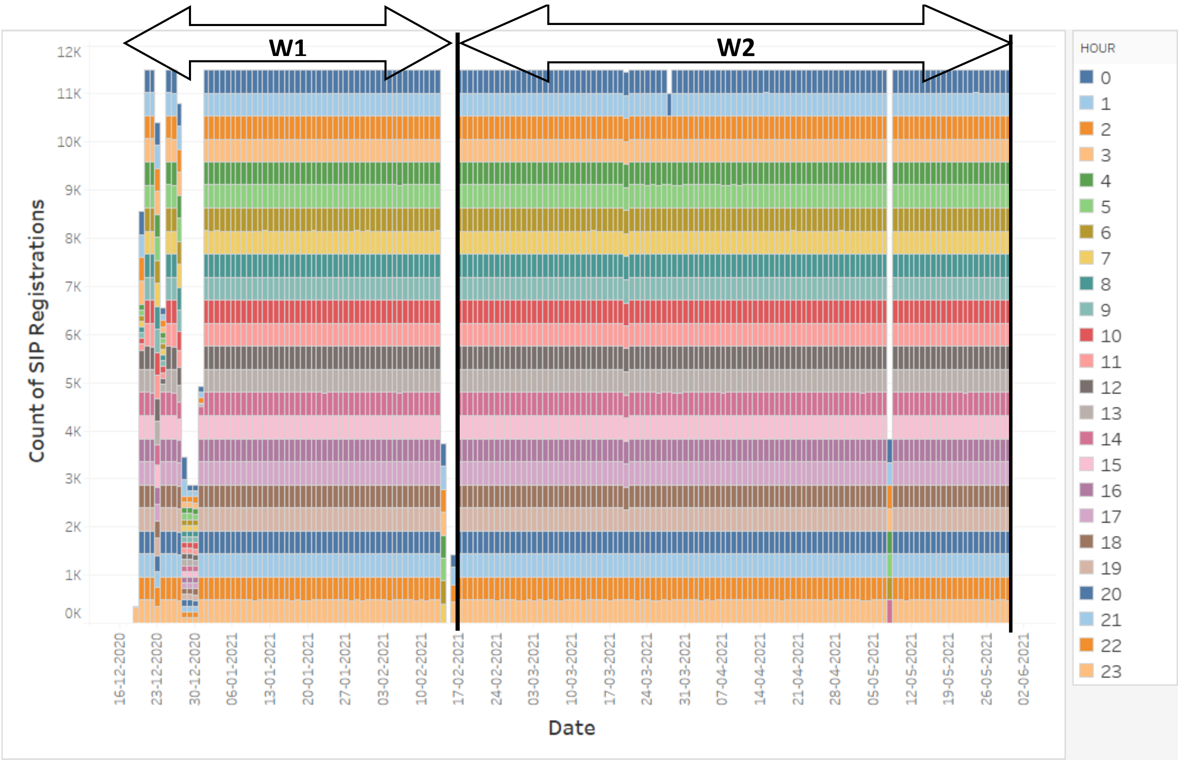



**Figure 6.4: Graphic showing distribution of SIP Registration events across days and hours on the 4 Raspberry Pi devices**

To test reboots, the Raspberry Pi reboots every night, a legacy intervention from when I relied on Twinkle and was observed to always register successfully. Prolonged outage of the cloud soft PBX was simulated by blocking access from the Raspberry Pis using Fail2ban® on the cloud server. When the blockade was lifted, the Raspberry Pis immediately registered successfully with the soft PBX. It was therefore shown that the SIP implementation can recover from outages that would be typical in distanced regions.

### 6.5.4 Performance of the Synchronization Service

The synchronization service is crucial to the operation of the RootIO software stack at the transmission site. It updates the cloud platform with the status of the device at the transmission site, while also downloading important configuration information necessary to achieve the desired broadcast setup at the transmission site. As such, a failure of the synchronization service in essence results in an estranged station and from working with communities in rural Romania, this necessitates manual intervention to restart the service, or even the device. The Raspberry Pi again benefitted from the lack of background Internet and power optimizations that plagued the Android mobile phone.

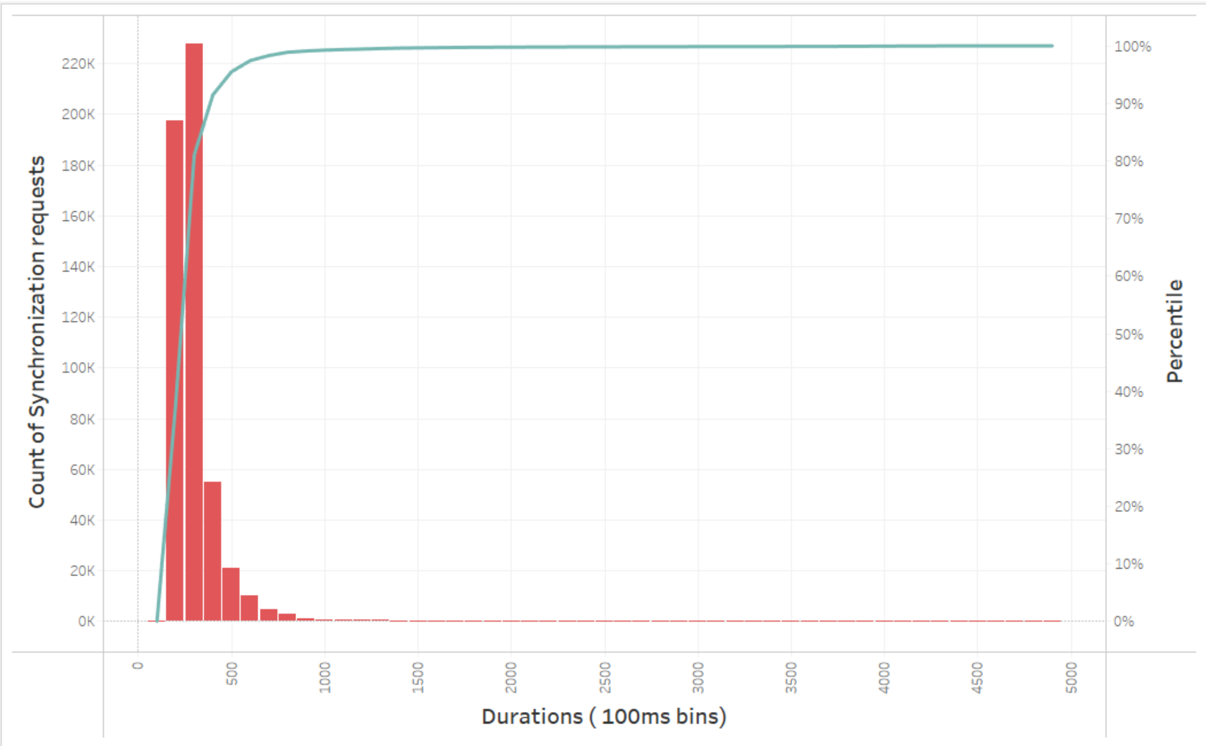



**Figure 6.5: Graph showing distribution (left axis) and percentile (right axis) of synchronization durations on the 4 Raspberry pis with consideration of requests within 5000ms (overall 95th percentile was 654ms)**

On the Raspberry Pi the synchronization service was assessed by analyzing 11,488,406 data points created by the synchronization service running on the four devices. Similar to analyses of the SIP and Radio services, there was demonstrated consistency over different hours during the day, showing that the service was always running, An analysis of the duration of the sync requests to the cloud HTTP API revealed an average duration of 376ms and a median duration of 275ms with the 95$^{th}$ percentile at 654ms. Excluding outliers due to server overload that resulted in slow processing, the synchronization service performed optimally. Similar to the performance observed in the stations in Romania, the modal bin for the 4 Rasspberry Pis was 300ms, suggesting that this is the expected duration.

| Window | Average (ms) | Standard Deviation | Median | 95th Percentile |
|---|---|---|---|---|
| 1 | 371.69 | 4705.21 | 259 | 785 |
| 2 | 340.17 | 3316.93 | 251 | 594 |
| 3 | 416.42 | 149911.54 | 272 | 542 |
| 4 | 366.28 | 375.8 | 327 | 583 |

**Table 6.1: Synchronization duration characteristics during the 4 observation windows**

An optimization that saw the synchronization service perform better on the Raspberry Pi, as can be seen in Table 6.1, had to do with the way that media is cataloged on the cloud platform. In prior approaches, the synchronization service always uploaded the entire media catalog, which the cloud platform would organize by artist, album, and eventually individual songs. This would then get incorporated into the existing catalog, taking care to avoid duplicates. Needless to say, this process was demanding on the server side and took a long time.

A latter implementation on the Raspberry Pi means that the synchronization of the media library is performed only when there is a discrepancy with the status in the cloud server (difference in counts of songs, artists, and songs), and a new synchronization is not performed while the server advises that a previous one is still ongoing. As such, the average synchronization duration is seen to be considerably lower for the Raspberry Pi. This resulted in significant performance gains compared to the previous window.

### 6.5.5 Performance of the Diagnostics service

The diagnostics service on the Raspberry Pi showed a steady performance and provided much more realistic performance statistics. An analysis of 491,532 datapoints relating to the collection of diagnostics revealed that the service was running at all times. As can be seen in Figure 6.6, by the fourth window of observation, the diagnostics service was seen to be running consistently, with the only dip observed on the 28th of March due to the start of daylight saving time.

Diagnostics relating to the performance of the actual Raspberry Pi such as memory, processor and storage utilization were easy to obtain reliably using standard Linux commands and APIs. For purposes of portability, however, I relied on the SystemInfo Java library from the Operating System Hardware Information (OSHI) Java project (https://github.com/oshi).

The acquisition of network metrics was, however, not as straightforward. There was no platform-independent way to measure the strength of Wi-Fi networks, and therefore I had to invoke shell commands depending on the platform. Wi-Fi strength on the Raspberry Pi was measured by executing the shell command "`iw wlan0 link`" and parsing the result for Wi-Fi strength information in decibels.

The Raspberry Pi also has the capability to use wired networks, but it was not clear how to measure their quality of service. This was, however, not a great concern, given that a credible measure of connectivity was the Raspberry Pi's success at synchronizing with the cloud sever, and, if it was able to do so, the rate of data transfer.

Regarding telephony diagnostics, these were obtained from the SIM 7600X 4G HAT by sending AT commands to measure the quality and type of the telephony network. This yielded granular information on the networking technology (variants of 2G/3G/4G), and the RSSI of the connected network. The SIM 7600X 4G HAT, as previously mentioned, was also capable of delivering accurate GPS data regarding

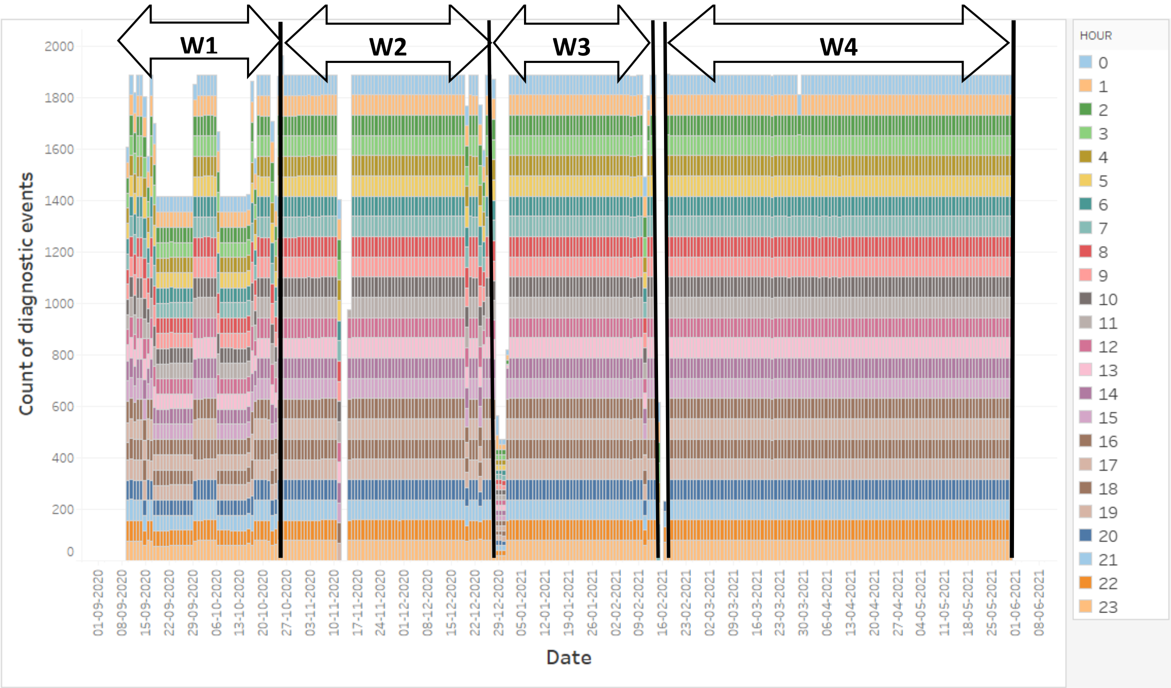



**Figure 6.6: Graphic showing performance of the diagnostics service across the 4 observation windows**

the location of the Raspberry Pi to which it was attached. GPS information was obtained by issuing an AT command that turned on streaming of location periodically, upon which an event would be triggered to save the received location information. I observed the modem over two windows to monitor how it performed and to detect any loss of connectivity with the Raspberry Pi. It was observed to always be reachable and connected to a telephony network on both 3G and 4G technologies as can be seen in Figure 6.6.

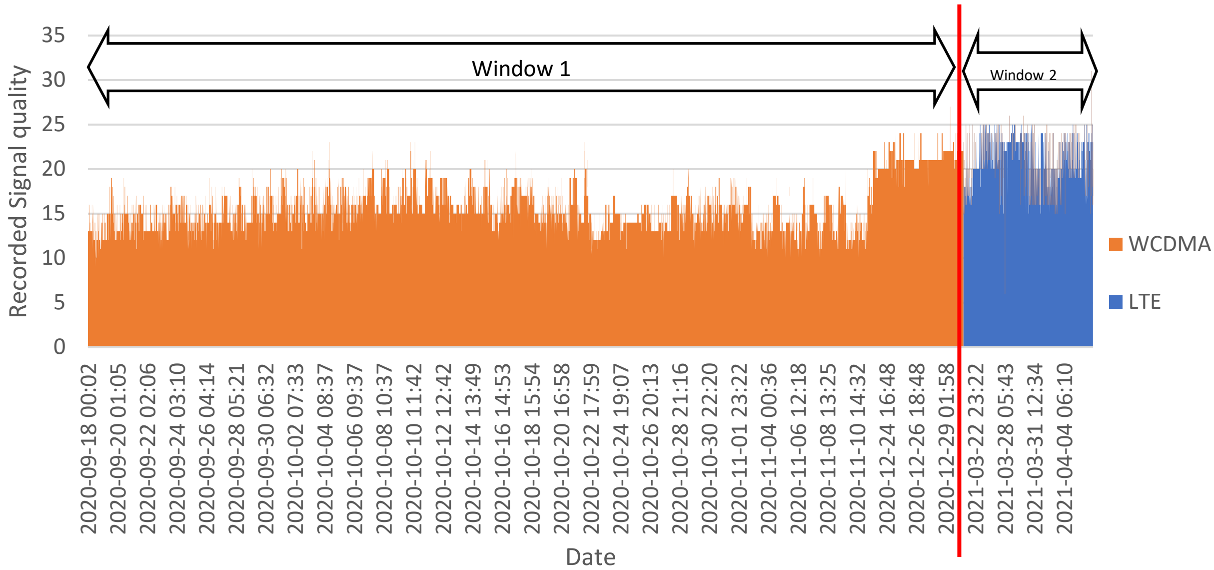



**Figure 6.7: Graphic showing network access for the SIM 7600X 4G HAT over 2 observation windows**

The one diagnostic that was not collectable was the battery status of the Raspberry Pi, for the simple reason that it does not possess a battery and is run directly from a DC input. On the mobile phone, a declining battery percentage was indicative of power outage at the transmission site and would help expose potential transmission outages. It is not impossible to add a battery to the setup, but some extra work would have to be incurred to obtain metrics about the power source. Overall, the metrics obtained give a rich overview of the performance of the technical components at the transmission site, the purpose for which the diagnostics service was implemented.

## 6.5.6 Performance of the Media Indexing service

The media indexing service was necessitated by the lack of automatic media indexing on the Raspberry Pi. This service does little other than periodically rebuild the media catalog by recursively traversing directories specified as containing media in the configuration file of the RootIO application. The observed performance of this service was satisfactory even without the optimizations that I mentioned in chapter 6.4.3. The media indexing service is supposed to guarantee two functions: That media indexing is run when there is no media in the catalog, and that the catalog is rebuilt when it is older than a configurable number of days.

| Songs Added | Timestamp |
|---|---|
| 1 | 2021-02-12 14:00:29 |
| 9 | 2021-02-12 14:00:31 |
| 29 | 2021-02-12 14:00:33 |
| 10 | 2021-02-12 14:00:34 |
| 1 | 2021-02-12 14:00:35 |
| 5 | 2021-02-12 14:00:36 |
| 16 | 2021-02-12 14:00:37 |
| 14 | 2021-02-12 14:00:38 |
| 16 | 2021-02-12 14:00:40 |
| 12 | 2021-02-12 14:00:41 |
| 22 | 2021-02-12 14:00:43 |
| 19 | 2021-02-12 14:00:44 |

```
|          27 | 2021-02-12 14:00:46 |
|          33 | 2021-02-12 14:00:48 |
|          12 | 2021-02-12 14:00:49 |
|          22 | 2021-02-12 14:00:50 |
|          22 | 2021-02-12 14:00:51 |
|           7 | 2021-02-12 14:00:52 |
|          15 | 2021-02-12 14:00:53 |
|          70 | 2021-02-12 14:00:56 |
|          20 | 2021-02-12 14:00:57 |
|         156 | 2021-02-12 14:01:01 |
|          21 | 2021-02-12 14:01:02 |
|          31 | 2021-02-12 14:01:03 |
|          36 | 2021-02-12 14:01:04 |
|          13 | 2021-02-12 14:01:05 |
|          12 | 2021-02-12 14:01:06 |
|          36 | 2021-02-12 14:01:07 |
|          11 | 2021-02-12 14:01:08 |
|          22 | 2021-02-12 14:01:09 |
|          31 | 2021-02-12 14:01:10 |
|           8 | 2021-02-12 14:01:12 |
|          15 | 2021-02-12 14:01:13 |
|          24 | 2021-02-12 14:01:14 |
|          38 | 2021-02-12 14:01:15 |
|         175 | 2021-02-12 14:01:16 |
+-------------+---------------------+
```

**Figure 6.8: Excerpt from MySQL database showing the performance of the indexing service on one of the Raspberry Pis**

It was observed that the media Indexing service was capable of indexing up to 1011 songs in a period of 37 seconds. However, it is hard to treat this as a rate given that this process is highly parallelized as was noted in chapter 6.4.3. The results show that the indexing is completed in an acceptable time given the large number of files, and that the cataloging is always done within the configured period. With the optimizations mentioned in chapter 6.4.3, it is possible to update the catalog much more frequently, all the while incurring a minimal performance penalty.

## 6.6 Optimizations on the cloud platform

The cloud platform was also optimized to guarantee increased uptime and minimize service disruption. During the time that I observed stations deployed in Romania, outages were observed on the cloud platform that hindered its use and performance. The advantage with the cloud platform is that it is easily accessible, and any rectification measures are instantly applied, as opposed to transmission site stacks that required physical intervention in most cases. I highlight the improvements made to the cloud platform and how these have eventually raised the performance of the platform, improved the general user experience, and guaranteed continuity.

### 6.6.1 Guaranteeing telephony service availability

The soft PBX that I was using was prone to failures to make calls, or programs failing shortly after the program for which they were initiated had launched. One of the reasons for this was that the soft PBX was the target of brute force attempts by malicious agents that try to exploit cloud PBX settings for free calls [34]. This often resulted in a denial of service that disrupted programs. These attempts also resulted in a severe increase in logging that filled up otherwise useful filesystems. This was addressed by installing and configuring Fail2ban, an intrusion prevention tool that could identify IP addresses matching brute force patterns in the soft PBX logs and block them.

Another issue that was frequently encountered was the failure of PSTN networks to connect calls initiated by the cloud server. These stations were, however, connected to the Internet and could be called directly via SIP. The improvement made here was to always try and call stations via SIP and fall back to PSTN networks only if the SIP call failed. This resulted in increased success in the rural Romanian communities given that it was easier to guarantee Internet to individual stations – whether via a SIM card or Wi-Fi router – than to guarantee a functioning trunk between our soft PBX and the telephony providers we were using in Romania.

Though they do not reside in the cloud, GOIP devices were introduced as an extra backup option when it was ultimately necessary to connect to the stations via PSTN networks. The advantage with GOIP devices is that control is entirely down to the operators of the station; they come with numerous drawbacks, however, as highlighted in chapter 6.4.1. Overall, their inclusion increased the chances of having a successful telephony connection to the transmission site.

### 6.6.2 Improving the web portal user experience

The web portal is the window for administrative users into the RootIO platform. From scheduling and uploading content, to monitoring the performance of stations, users in most cases only need to interact with the portal to perform all the routine activities necessary to keep their RootIO stations operational. The most common issues that affected the operations of the web portal included failure of the entire stack because of the unavailability of space following SIP attacks that resulted in excessive logging. To mitigate this issue, separate drives were used for media, log files, and database files.

The site was also originally implemented only in English with little support for localization. To facilitate this, I put in place an automated pipeline that uploads all the phrases on the site to a translation site (https://www.transifex.com). Users can then translate the RootIO portal into a language of their choice, and these translations are updated in the RootIO cloud on a nightly basis. This made it possible for volunteers to translate the site into Portuguese and Romanian in the Grassroots Wavelengths project that relied on the RootIO platform to set up community FM radio stations.

Lastly, some outages were due to issues such as expiry of the domain name of the portal, which was fixed by establishing proper routines to proactively mitigate this potential occurrence. Another proactive step was the integration of site monitoring and alerting using Site 24x7 (https://www.site24x7.com), as well as setting up an Elastic® stack monitoring dashboard that collects logs to show performance of the platform as a whole.

There were also a number of usability issues reported by users of the RootIO web platform indicating areas where better design could make the website more intuitive. It is worth noting that my research was centered on creating a minimal stack that allows for communities to take on FM broadcast. My work is novel in many areas, including in UI/UX design for an intuitive interface that obviates the possibilities afforded by the RootIO system. A quick solution in this case was to institute a Wiki that details the operation of the platform as it currently is, before making interventions that make it more intuitive for grassroots communities.

## 6.7 Considerations for coverage visualization

As the reader will remember, the very first deployments of the RootIO stack were geared at evaluating whether FM broadcast was achievable to a quality indistinguishable from that of prevailing commercial FM stations, all while using a minimal software and hardware stack. Through successive deployments, a model was arrived at that addresses both the technical and social considerations for sustainable FM broadcast in distanced regions. In this section, I detail my work towards SPLAT4J, an open source Java rendition of a popular coverage mapping tool SPLAT [71].

In all the communities where I worked, there was no knowledge of RF electronics, meaning that communities had no way of assessing what factors influence the coverage of their desired station, or of visualizing the impact of these factors. This information was, however, always required by communications regulators ahead of issuing licenses. SPLAT4J visualizations would allow a community to:

- Determine the optimal location to locate the antenna to achieve maximum coverage.
- Identify the ideal equipment based on a simulation of the coverage used by different parameter configurations.
- Identify how many transmission points can be collocated in a region on the same license without interference.

It is worth noting that tools such as SPLAT4J, though providing clear indications as to the correct procedures, are nothing more than models. A study also showed that the ITWOM model, on which SPLAT4J is based, was much more negative in some circumstances [60], but these are not applicable to RootIO stations given their limited reach. It is practically impossible to cater for all the dynamic factors that influence the received FM signal at a particular point, but one can model the relative FM signal qualities at different locations with relation to the key factors: ERP, antenna altitude, frequency, and topography. The choice of these factors is fortunately within the control of the community and can therefore help the community to plan for ideal coverage. My contributions in SPLAT4J over SPLAT are:

- Porting SPLAT to Java, a more platform independent language, allowing it to run on virtually all imaginable computing platforms.

- Improving the generated image format to PNG format as opposed to PPM format, resulting in images reducing in size by a factor of 10.
- Provision for the generation of translucent images, allowing for the overlay of the resulting images on maps to show coverage in a more relatable manner.

## 6.7.1 Propagation estimation

There are a number of RF coverage modelling formulas, but the most used and accessible are those based on Longley-Rice calculations. Two widely used formulas derived from Longley-Rice calculations [69] are the Irregular Terrain Model (ITM) and the Irregular Terrain with Obstructions Model (ITWOM). ITWOM replaces and supplements obsolete diffraction calculations with RTE functions, allowing for more realistic modeling given consideration for obstructive clutter that plays a major role in RF propagation by absorption, modeled on Beer's law. Results from a comparison of both formulas show that the ITWOM formula is more accurate over distances of less than 20km, but less accurate for distances over that limit [60].

I therefore based my research on ITWOM, given the short distances covered by RootIO stations, as well as the low antenna heights that result in increased relevance of the degradation due to ground clutter and diffraction. ITWOM is relied on in modeling that is implemented in the SPLAT, and subsequently SPLAT4J modeling tools. The reader should be reminded that ITWOM is a probabilistic estimation approach for the received signal strength over a large area and would be representative of a large area over a long period of time, rather than an individual receiver instantaneously.

The terrain data that I used in my modeling is obtained from the database of the SRTM project [109]. The supplied terrain information has a pixel size of 25M at its highest resolution and 90M at the lowest, giving a highly accurate representation of the earth. While the formulae applied allow for the inclusion of ground clutter, the communities I worked in in Northern Uganda were rural and characterized by savannah ecosystems, thereby presenting minimal ground clutter. As such, the most influential factors in RF modelling were the ERP, transmitter height and receiver height. Below I detail the factors that influence RF propagation, their impact, and their consideration in a tool for coverage visualization.

## 6.7.2 RF factors and their impact

To arrive at the eventual considerations used in modeling, we start by recalling that the power density present at any point from an isotropic antenna is given by:

$$I = \frac{P_t}{4\pi d^2} \tag{6.1}$$

Where Pt is the transmitter power, and d is the distance from the transmitter antenna. The Power received by a terminal is however dependent on the efficiency of the receiving antenna thus:

$$Pr = IA_{eff} \tag{6.2}$$

Where the receiver antenna efficiency is given by:

$$A_{eff} = \frac{\lambda^2}{4\pi} \tag{6.3}$$

Where λ is the wavelength. From the above equations:

$$P_r = \left(\frac{P_t}{4\pi d^2}\right)\left(\frac{\lambda^2}{4\pi}\right) \tag{6.4}$$

The formula for the free space path loss then materializes thus:

$$L_p = \frac{P_t}{P_r} = \left(\frac{4\pi d}{\lambda}\right)^2 \tag{6.5}$$

Given $f = \frac{c}{\lambda}$ Where c is the speed of light:

$$L_p = \left(\frac{4\pi df}{c}\right)^2 \tag{6.6}$$

To achieve the convenience that comes with representing this loss in dB:

$$L_p = 10\, n\log_{10}\left(\left(\frac{4\pi df}{c}\right)^2\right) \tag{6.7}$$

Therefore:

$$L_p = 32.44 + 10n\log_{10} f + 10\, n\log_{10} d \tag{6.8}$$

Where: ***f*** is the frequency in megahertz, ***n*** is the path loss exponent (with value 2 for free space) and ***d*** is the distance between the transmitting antenna and the receiving antenna in kilometers. Further, the maximum propagation distance for a signal from an antenna at height h meters above ground level, assuming no obstructions is approximated by:

$$d_m = 4.12\sqrt{h} \tag{6.9}$$

Equation 6.8 represents the free space path loss, which is the attenuation of radio waves before factoring in environmental conditions. Other environmental factors and ground clutter contribute to the overall loss resulting in a representation of overall attenuation as:

$$A_{TOT} = (A_{RTE-AC}\ or\ A_{RTE-BC}\ or\ A_{DIFF}\ ) + \ A_{2R} + \ L_p \tag{6.10}$$

Where $A_{DIFF}$ is the attenuation beyond the horizon due to diffraction, $A_{RTE-AC}$ is the attenuation above the clutter canopy, $A_{RTE-BC}$ is the attenuation below the canopy and $A_{2R}$ is Two-Ray multipath attenuation. However, for the frequency range, antenna height and transmission distances used in the RootIO setups, multipath attenuation is of limited significance when compared to other loss terms and is shown to have little effect in actual measurements [100]. Any other environmentally dependent terms are minimized within the limited range of RootIO stations with a targeted 15KM radius audience. Below I briefly recap the factors considered in propagation modeling and their impact.

**ERP:** This is the power emitted by the transmission assembly. It is a product of the power emitted by the transmitter in decibels, minus losses due to attenuation in the RF cable, plus the gain of the transmitting antenna in decibels. The loss obtained from equation 6.1 above is directly subtracted from the ERP giving the received signal strength at a particular distance from the transmitter.

**Transmitter Antenna height:** The height of the antenna serves first to increase the distance reachable by the emitted signal. As can be seen from equation 6.9, increasing the height increases the covered area. A higher antenna also effectively leads to an elevated Fresnel zone, resulting in less interference by ground clutter.

**Receiver Antenna height:** The receiver antenna height serves the same purposes as the transmitter antenna height, increasing the reachable distance as well as clearing clutter that would pose obstacles in the Fresnel zone. For purposes of this simulation, the consideration is that receiver antennae are at 10m above ground, in line with similar consideration as measurements performed by OFCOM [81] to determine useable coverage levels for FM signal.

**Radio Climate:** Climate determines the characteristics of the medium through which radio waves travel. Temperature, pressure, and humidity all affect the propagation of radio waves, and are catered for in the probabilistic modeling relied on by ITWOM, and inherently also relied on by SPLAT.

**Frequency:** As was seen from equation 6.1, loss is logarithmically proportional to the frequency. This results in a marginal disadvantage for higher frequencies. It should be noted that this loss is not due to the frequency but rather the efficiency of the receiving antenna.

**Earth Dielectric constant:** For emissions below 30MHZ, the main medium of propagation is ground waves; even though this is considered in ITWOM analysis, it is of little consequence for frequencies in the UHF range, such as those used in FM radio and with transmitting antennae 5 meters above ground level.

**Earth Conductivity:** The path from the transmitting antenna to the receiving antenna is rarely free of obstacles. These reflect radio waves with varying efficiency, but generally the higher the conductivity, the better the reflection achieved. The effect of conductivity is also minimal in the VHF frequency range and with transmitting antennae more than 5 meters above ground level.

**Atmospheric bending:** Radio waves travelling through the atmosphere are subject to refraction in accordance with Snell's law. Changes in the refractive indexes of adjacent media result in the waves being refracted. Given the higher refractive index of air near the earth's surface, it follows that radio waves are refracted towards the earth as they travel through the atmosphere, allowing for further reach than is allowed by line-of-sight.

**Polarization:** Though considered for analysis, the polarization of the antenna has little bearing on the field strength received at a particular location. All the setups of RootIO stations used horizontally polarized antennas. Measurements have also shown insignificant variations between setups using both polarizations [69].

**Fraction of situations:** This is a measure of the expectation of the received signal strength at a particular location. For reasons such as measurement errors, human errors and the availability of someone to take the measurement, the received signal strength at a given location cannot be perfectly determined. This measure therefore stipulates the minimum fraction of measurements for which the reported maximum signal strength should be observed.

**Fraction of time:** This is the fraction of time for which the measured signal at a given location should be greater or equal to a specified threshold. In simpler terms, the reported signal strength is the strongest value measured for the specified fraction of time.

**Ground clutter:** Objects at the surface of the earth reflect, refract, and absorb electromagnetic radiation. The more clutter there is in the immediate environment of the receiving antenna, the greater the degradation of the received broadcast. Consideration for ground clutter is one of the main drivers for the move to the improved ITWOM modelling for RF propagation.

## 6.7.3 Variable settings for site analysis

As priorly mentioned, the most important factors in modelling the coverage of stations are the transmitting antenna height, frequency, receiver antenna height and ERP. The rest of the factors cannot be changed by the community and are known for large geographical areas: in many cases choosing a country from a drop-down list is enough to set the radio climate, ground conductivity and earth dielectric constant. Factors such as the clutter and receiver antenna height can be left to users, but these might be hard to interpret and could be encapsulated in an "urban/rural" choice. Variability factors can be masked in a "reliability" scale if left to the choice of users, though in my modeling I use the standard (50, 90).

Below are the factors that I used to model the coverage of the various sites at which I deployed RootIO stations.

| Factor | Pabbo | Patongo | Varvoru d Jos | Sfantul Gheorghe |
|---|---|---|---|---|
| ERP (W) | 33.4 | 24.9 | 66.7 | 66.7 |
| Frequency (MHz) | 103.8 | 103.8 | 103.8 | 106 |
| Antenna height | 15M | 15M | 20M | 20M |
| Receiver antenna height | 10M | 10M | 10M | 10M |
| Maximum radius | 15KM | 15KM | 15KM | 15KM |
| Earth Conductivity | 0.005 $SM^{-1}$ | 0.005 $SM^{-1}$ | 0.005 $SM^{-1}$ | 0.005 $SM^{-1}$ |
| Earth Dielectric constant | 15 | 15 | 15 | 15 |
| Atmospheric Bending Constant | 301 N-units | 301 N-units | 301 N-units | 301 N-units |
| Polarization | 0 (Horizontal) | 0 (Horizontal) | 0 (Horizontal) | 0 (Horizontal) |
| Radio Climate | 1 (Equatorial) | 1 (Equatorial) | 6 (Maritime Temperate, Over Land) | 6 (Maritime Temperate, Over Land) |
| Fraction of situations | 0.5 | 0.5 | 0.5 | 0.5 |
| Fraction of Time | 0.9 | 0.9 | 0.9 | 0.9 |
| Minimal signal strength | 48 dBµV/m | 48 dBµV/m | 48 dBµV/m | 48 dBµV/m |

**Table 6.2: Values used for RF propagation modelling for existing transmission sites in Romania and Uganda**

The frequency and antenna height values were determined by the licensing authorities in both Uganda and Romania. The minimal signal strength and receiver antenna height were taken from OFCOM recommendations for useable FM coverage which call for 48 dBµV/m for mono audio with receiving antenna height assumed at 10 meters above ground [81]. Modelling literature suggests common values for the atmospheric bending constant as 301 N units, an earth dielectric constant of 15 and earth conductivity of 0.005 SM-1 [9,70]. The antennas used in all deployments featured horizontal polarization and their gain was factored into the ERP. The resultant simulation is desired to hold for at least 50% of the considered area, for 90% of all times, hence the 0.5 and 0.9 fractions of situations and time, respectively. Radio climate values were taken from the ITU recommendation p.2001-3 [140].

Of all the variables taken into consideration, antenna height and ERP are the two that communities are able to manipulate. I begin by modelling the coverage of the existing installation at each of the locations for which maps are included in Annex D: Coverage models of different transmission sites modelled by SPLAT4J. I then show what the coverage would look like if one were to change antenna height or ERP. The models revealed that antenna height was the biggest determinant for coverage. At similar heights in 3 communities whose coverage maps are included in 11, it is observed that the usable coverage varies infinitesimally with significant increase of ERP.

Sfantul Gheorge was an ideal location to observe the two variables at play given its entirely flat landscape which resulted in a perfect coverage circle. While increasing the ERP had no effect on the area covered by usable broadcast (Figure 11.9 and Figure 11.10), increasing the height by 30M resulted in an increase of 3KM in the coverage radius (Figure 11.9 and Figure 11.11). However, it would be a logistical challenge for communities to erect towers up to 50M in height, and collaborations should therefore be sought with telecommunications providers to leverage their antenna deployments that are capable of achieving these heights. The gains mentioned in 6.5 allow for deployments at ideal but hard-to-reach areas given the reduced need for human intervention.

Tools for modelling coverage mapping should therefore emphasize transmission antenna height given the insignificant contributions of other factors such as frequency and ERP at the scale of stations proposed by RootIO. Finally, it was realized that a practical radius for RootIO stations is 10KM rather than the previously sought 15KM. This, however, is not a problem, given that from conception, the idea of RootIO is that transmission points multiply rather than expand.

# Chapter 7

## Conclusion

In this chapter, I lay out the assumptions that may present significant challenges to researchers seeking to reproduce or build on my work. I give tips on how to circumvent these challenges, as well as highlighting interesting areas of research that would further advance the thesis of my research.

The reader by now has a good understanding of a stack that hands control of FM radio, an ubiquitous technology, to communities that depend on it the most. As the developed world moves into digital technologies, there is little incentive besides charity to invest development resources into catering for the needs of the least financially appealing populations. It is therefore not only prudent, but also possible to package the benefits of digital advancement into solutions, some of them analog, that make sense to communities that are underrepresented in contemporary technology discussions.

FM radio is not a technology that distanced communities are prepared to manage in its present form and stripping out components does not necessarily arrive at a model that is accommodable within distanced communities. It is not clear if this is a cause or result of the hesitance by regulatory authorities to hand control of this space to smaller community-sized entities. My work however goes a long way in breaking this deadlock, resulting in easily manageable FM setups that are acceptable to licensing entities in different countries. I go on to show that these FM setups work for these communities and that these communities have in turn owned them and customized them – because for the first time, they can.

RootIO is shaped and trimmed by numerous communities, resulting in something very basic that allows all communities to initiate the discussion on FM radio ownership with regulators. Subsequently, communities learn the principles of FM radio and superimpose their own societal realities, resulting in FM radio that is tailor-made for individual communities. From a higher perspective, one could say that this is the materialization of the Internet of FM radio.

## 7.1 Risks and assumptions

My thesis takes on an area that is uncharted and builds a framework for FM radio in communities that are the most distanced from modern technologies. Ultimately, I was able to show that the technology works, that societies have been able to own it and customize it, and that regulators see it as a stack worthy of licensing. This however has not been free of risks and in some cases, the people that I have worked with, coupled with my background, have obviated some of the pitfalls that lie in wait of anyone seeking to build on this work.

Below, I would like to highlight some of the assumptions that I made, but which are founded on discussions and observations while working with the different communities in which I installed and studied RootIO setups. I also touch on why they are realistic and show that in at least one instance, they hold true and therefore are not merely utopian.

### 7.1.1 Licensing regimes

From my observation, different countries had different licensing regimes. Some countries were more advanced than others when it came to accommodating community broadcasting – at least on paper. Regardless, no country had well-defined permanent licensing for stations at the scale proposed by

RootIO. As such, licenses were in many cases a loose fit and involved discussions between the regulator and a reputable entity in that community, meaning the RootIO technology is not yet a turnkey FM radio setup solution.

The process of applying for FM licenses came with varying levels of complexity, but in all contexts was intractable for grassroots communities. In Romania, Ireland and Uganda, the regulator demanded an articulate business plan on top of detailed technical information. In all these settings, the communities sought the help of FM radio technicians and media professionals. The possibilities of a remote community obtaining a license are therefore contingent on their access to media and technical professionals willing to assist with the process of applying for a license.

However, in all communities where licenses were applied for using RootIO as the basis platform, FM licenses were granted, and in some cases the number of licenses was increased over time. My design is also pragmatic, one of its key design considerations being the possibility for a lobbying entity to obtain and hold a license on behalf of communities that cannot tract the licensing process. This, indeed, was the model in Uganda and Romania, where RootIO stations were operated in communities for an extended period of time with a company holding the licenses on behalf of the communities.

As a community telecommunications operator mentioned to me, it is a common mistake by practitioners to assume that licensing entities are inclined to deny license requests for political and other motives, unaware that in many cases regulation lags behind innovation and novel interventions are ill-accommodated by existing regimes. Regulators are more likely to create problems for themselves by granting requests rather than denying them, and, as such, efforts should be made to engage regulators in dialogue based on evidence, and to demonstrate consideration for concerns in the area of interest.

### 7.1.2 Varying telephony landscapes in different countries

The RootIO setup depends heavily on telephony both for human-to-machine and machine-to-machine communication. During the time in which I conducted my research, I integrated with telecommunications providers using a range of options. First, I integrated with a telecommunications provider in Uganda for whom I had worked previously. This service however was not being offered by them: I obtained it only from prior knowledge of its existence.

The aforementioned service was to a fixed switch and did not provide some of the advantages of mobile telephony such as SMS messaging. I subsequently integrated with mobile networks using a GOIP device as explained in chapter 3.3.1. This however meant that I had a hardware device that needed hosting, a disadvantage given that the rest of the RootIO stack was in the cloud. Furthermore, prepaid SIM cards such as the ones that we were using in the GOIP device need constant maintenance in the form of topping up and monitoring network status. Fortunately, this was automatable, but created extra work.

In European contexts, it was possible to interconnect with aggregators that provided connections to different countries. The quality of the connection was, however, down to the terminating telecommunication provider, and was not always guaranteed. In my subsequent work in the telecommunications industry, I came to understand that monitoring the quality of telephony routes provided by such providers is a full-time job, and requires constant shuffling between providers when quality dips are observed.

The introduction of direct SIP telephony support in the RootIO applications on the Android mobile phone and Raspberry Pi pushed this problem to the background, given the stellar performance of this setup. However, access to usable Internet is not guaranteed in most of the communities that would benefit the most from RootIO. The telephony options detailed in my research work with varying levels of success, but this is ultimately down to having access to a person knowledgeable in the telecommunications area.

### 7.1.3 A need for basic literacy

The RootIO idea was conceived to fit in communities that feature some of the least desirable socioeconomic demographics, one of them being reduced literacy. The emphasis on audio technologies should galvanize the resulting solution against this challenge but setting up and administering RootIO stations requires a fair amount of literacy, both basic literacy and computer literacy, as well as access to a computer or tablet with Internet connectivity. It is therefore necessary to have someone in the community with these skills and the willingness to assist the community.

It has been my observation that these kinds of people are not impossible to come by, even though they are in short supply in distanced communities. In Northern Uganda I installed the stations in community ICT centers installed by UNICEF and naturally these were places where people with literacy skills and computers resided. This however did not always result in ideal service, as in some cases these ICT centers were collocated with other organizations that incorporated them into their own agenda and left little space for community participation.

My work in northern Uganda took place in areas that had been recovering from war and were therefore attracting many NGOs involved in rehabilitation efforts. The NGOs greatly skewed interests towards a financial dimension but also utilized whatever few capable volunteers there were in these communities. These volunteers were therefore in high demand from many development projects in the area. This meant that their dedication to the RootIO stations was divided, and they could not focus on maximizing the benefits of RootIO within their communities. It is therefore emphasized that RootIO stations should be commercially viable, and this possibility has been demonstrated. This enables the stations to have full-time personnel who seek and share in the financial success of the station.

### 7.1.4 Access to equipment

RootIO uses devices that are not easily accessible in distanced regions. Even in European contexts, some of the devices necessary for the RootIO stack need to be ordered from other countries, and can take up to a month to be delivered. This issue is compounded for distanced regions, where delivery networks may not exist, and online purchase platforms are not usable due to lack of supported payment methods and Internet. The Raspberry Pi, FM transmitters and antennas, 50Ω impedance outdoor RF cable, and connectors are all things that were impossible to find even in the capital of Uganda, and therefore had to be imported.

A proposal was therefore for the RootIO project to sell or assemble kits on order, as has previously been y done for deployments in parts of Europe. It is assumed that communities in these distanced communities that want to use RootIO technology have contacts with organizations that can assist with the procurement and delivery processes or RootIO equipment. In the communities that I have worked with in both Northern Uganda and Romania, there were ties to entities capable of sourcing and delivering components unavailable within the community.

### 7.1.5 Cost implications

The cost reductions in FM radio station setup and operation afforded by RootIO are evidenced in my thesis. For less than €5000, a community can set up their own low-power FM radio station. This is the price of two-three averagely priced laptop computers. In the communities where I worked in Northern Uganda, Romania and Ireland, I observed that there were investments in ICT equipment to a sum greater than this value. In Northern Uganda, this was mostly in the form of ICT centers donated by UNICEF and other NGOs. In Romania and Ireland, our community partners demonstrated availability of funds within their various programs to bankroll the setup of these stations.

However, it is a false assumption that the cost of these stations has been lowered to the point where communities can afford to set up these stations on their own. Many distanced communities have a lot of competing priorities, and may therefore find it hard to pool the financial resources necessary to set up RootIO stations. The stations' dependence on funding entities also leaves them vulnerable to the agendas of the funders, which may conflict with the communities' interests and beliefs.

### 7.1.6 Copyright issues – replaying content and streams and podcasts

In all the rural communities where I worked, I observed that people had access to music and content mostly by sharing audio files in ways that do not comply with copyright regulations. Most of the stations did not have in place a policy for copyright compliance and enforcement was lacking by the relevant authorities. In Romania, the strict copyright requirements necessitated the implementation of the media log functionality in the RootIO platform, and this was used to compensate artists whose content was played on local stations.

Communities need to be fully aware of the liabilities they incur by broadcasting content that may be covered by copyright law, but unfortunately this is an area for which expertise is lacking in many distanced communities. There is no easy way to incorporate copyright checks into the RootIO platform besides relying on station administrators to schedule only content for which they have explicit permission to broadcast. It was also observed when sourcing content from streams and podcasts that different sources had different copyright terms, some more stringent than others. This gives more relevance to the role of content providers in the RootIO platform: they centralize copyright and other content issues so that individual communities do not have to concern themselves with such issues, but this role has been scarcely used in RootIO setups.

## 7.2 Future work

RootIO as an initiative is part of a field of research in which many other components are missing. For the full benefit of RootIO to be realized, it is necessary that work is done in other areas both to further the RootIO idea, but also to build upon similar innovations in other technologies used in distanced communities. Below I touch on some ideas that belong in the same field, and some of which leverage my research with promising results.

### 7.2.1 Community telecom initiatives

Community Telecom companies are able to do for the telecommunications field what RootIO has done for the FM field. This would bring another appropriately sized technology stack into these communities, allowing for new synergies to develop between FM radio and telecommunication companies. I have already mentioned in my thesis that working with large telecommunication companies was always a challenge for RootIO, as the economic opportunity was deemed infinitesimal. In a community telecommunication company, the economies of the radio stations and the telecommunications company would most likely line up, as can be seen in [48] .

Community telecom initiatives have the added disadvantage of operating in a frequency space that is highly profitable and is currently in high demand. It would, however, be ideal to deploy RootIO stations in communities that are already using community telecommunication companies, to see if a synergy can develop that is similar to the one found between TV, FM radio and telecom companies to form media conglomerates. The increased service footprint would give community telecom initiatives an increased chance of success in regulated spectrum space.

### 7.2.2 Language resources

Many communities in distanced regions display reduced literacy rates and speak languages that are not well represented in modern technologies. Given the lack of language resources for these communities, they are often excluded and must rely on English and a handful other languages that are widely supported in the digital space. Research towards generating resources for these communities would allow use of technologies such as TTS and speech recognition, both of which go a long way in increasing the participation of these communities.

In my work with communities in rural Romania, it was reported by the local communities that the TTS voice generated during previously mentioned research around RootIO [94] was identical to and mistaken for that of a local volunteer in the community, demonstrating how successful the end product was. This shows how sophisticated technologies can be situated with a seemingly antiquated technology to grow an information culture in distanced communities. Progress in speech recognition is not as advanced but would be extremely useful in RootIO settings. Currently, navigation of telephony menus is by DTMF digits, but this could be done by prompting for speech, a more natural alternative in oral communities.

### 7.2.3 Hardware integration and encapsulation

RootIO's hardware has been under constant evolution since the first prototype. In fact, the first transmitters in the prototype did not even make it to deployment in the communities. RootIO, however, has reached a point of maturity where a "Long Term Support" hardware stack is imaginable. It is known what computing devices, transmitter brands, cables and antennas are required for optimal performance. It is therefore prudent to consolidate these choices into integrated hardware and installation kits to save communities the labor and mitigate the possibility of errors incurred when interconnecting devices.

The one integration that offers the most gains is that between the Raspberry Pi and the FM transmitter board. This would allow for monitoring and manipulating transmission characteristics in the RootIO portal. At present, it is possible for members of the community to misconfigure their transmitters, as was witnessed in Northern Uganda, putting them at risk of non-compliance with the license for their station. Changing the frequency also risks broadcasting at a frequency other than the one for which the transmission line is optimized, resulting in potential damage to the transmitter and reduced coverage. Not only does integration reduce the number of devices to configure, but it also reduces the interfaces users have to deal with.

Another area to pursue is the visualization of the broadcast characteristics of the transmitter. In many cases in Northern Uganda, I observed that the transmitter reported less than ideal conditions for broadcast, but that the community could not interpret the results of this. Using the SPLAT4J tool that I ported to Java for cross platform use, it should be possible to feed the transmitter characteristics into a simulation, and then to display to the station administrators the current coverage of the station, as well as the performance of neighboring stations.

The deployment of RootIO stations in Cape Verde sought to use RTL SDR dongles as illustrated by Danymol et al [27] to measure the signal strength of the transmitter and feed this into the RootIO platform for increased visibility into the operation of the transmitter. This has the disadvantage that it cannot be entirely relied on to paint a complete picture of the transmitter performance as many deployments would be needed in different locations, with careful consideration for siting. It can, however, provide important metrics such as frequency, audio levels, audio mode (stereo or mono), and transmitter power, thereby providing a community-based sensor for evaluating the radio experience. In Romania this was achieved by merely feeding the audio from an FM receiver into a Raspberry Pi that would then stream the audio online. The assessment was subjective and entirely based on human judgment.

### 7.2.4 A model for a holding entity

An evident pattern in all the RootIO deployments was the need for an intermediate entity to assist communities in issues such as negotiating with licensing regulators, figuring out logistics for station installation, and ensuring that best practices are observed as stipulated by broadcasting authorities. In all the setups where this entity was present, RootIO stations registered better performance. It was observed that communities could not initiate the discussion with regulators, nor fully understand the stipulated regulatory requirements.

In Romania, the intermediate organizations also helped with media strategy and prioritized community requirements for implementation, either as add-ons or as core components of the RootIO platform. This was an important factor that allowed investment of development resources into components that had the most impact. From observation of the stations in Uganda and Romania, I conclude that this holding entity ideally should:

- Hold licenses for the different FM radio stations and, where possible, plan for the sharing of the license between different communities.
- Plan for and oversee training of personnel involved in operating RootIO stations.
- Ensure that regulatory requirements are observed, and put in place measures for monitoring and evaluating them.
- Identify opportunities and bottlenecks, and recommend ways to harness the former and eliminate the latter.

### 7.2.5 An entry point for AI

Analog FM radio is still going strong, even in the face of on-demand media platforms and the emergence of digital radio. By the end of my research, it was clear how advances in computing such as machine learning and artificial intelligence can be applied in the FM radio space in ways that benefit distanced communities. Given that RootIO automates a lot of the tasks around FM station management, it is possible to introduce AI into these processes, further reducing the need for human labor. Program scheduling and content sourcing are all prediction problems that are well taken care of in digital streaming platforms, and for which inference can be made from existing community selections to build dynamic programs.

Another area where machine learning can be introduced is in determining the best channel to reach the transmission sites. The devices at the transmission sites are constantly sending information about the networks they are connected to, as well as synchronizing with the cloud platform. This information can be relied on to dynamically classify stations into high and low bandwidth categories, as opposed to the current categorization, which is manual. This would allow for the best audio quality whenever possible.

# References


[1] Tonye Adikumo. 2017. Infrastructure: NCC secures N221 million waiver for MTN in Kano. *TODAY*. Retrieved July 24, 2017 from https://www.today.ng/business/telecoms/299514/infrastructure-ncc-secures-n221-waiver-mtn-kano

[2] Sheetal Agarwal, Arun Kumar, Amit Anil Nanavati, and Nitendra Rajput. 2008. Voikiosk: increasing reachability of kiosks in developing regions. In *Proceedings of the 17th international conference on World Wide Web*, ACM, 1123–1124.

[3] Mary Alexander. 2009. Google's SMS internet for Uganda. *Brand South Africa*. Retrieved September 19, 2017 from https://www.brandsouthafrica.com/investments-immigration/africanews/googles-sms-internet-for-uganda

[4] Najma Alghawi. 2019. A Study on SIM box or Interconnect Bypass Fraud. (February 2019). Retrieved May 22, 2021 from https://bspace.buid.ac.ae/handle/1234/1515

[5] Morgan G. Ames. 2019. *The charisma machine: the life, death, and legacy of One Laptop Per Child*. MIT Press, Cambridge, MA.

[6] Jens Christopher Andvig. 2000. Family-controlled child labor in Sub-Saharan Africa-A survey of research. (2000).

[7] Roger Baig, Felix Freitag, Agusti Moll, Leandro Navarro, Roger Pueyo, and Vladimir Vlassov. 2016. Cloud-based community services in community networks. In *2016 International Conference on Computing, Networking and Communications (ICNC)*, IEEE, Kauai, HI, 1–5. DOI:https://doi.org/10.1109/ICCNC.2016.7440621

[8] Rozália Klára Bakó. 2020. Romania: Right To Repair for Sustainable ICT Use. *Glob. Inf. Soc. Watch* (2020), 207–210.

[9] B. R. Bean and G. D. Thayer. 1959. Models of the Atmospheric Radio Refractive Index. *Proc. IRE* 47, 5 (1959), 740–755. DOI:https://doi.org/10.1109/JRPROC.1959.287242

[10] Alexander Bick, Adam Blandin, and Karel Mertens. 2021. Work from Home Before and After the COVID-19 Outbreak. *Fed. Reserve Bank Dallas Work. Pap.* 2020, 2017 (February 2021). DOI:https://doi.org/10.24149/wp2017r2

[11] Nicola J. Bidwell, Roberto Cibin, Conor Linehan, Laura Maye, and Sarah Robinson. 2021. Being Regulated: Licence to Imagine New Technology for Community Radio. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW1 (April 2021), 1–27. DOI:https://doi.org/10.1145/3449228

[12] Victor de Boer, Pieter De Leenheer, Anna Bon, Nana Baah Gyan, Chris van Aart, Christophe Guéret, Wendelien Tuyp, Stephane Boyera, Mary Allen, and Hans Akkermans. 2012. RadioMarché: distributed voice-and web-interfaced market information systems under rural conditions. In *International Conference on Advanced Information Systems Engineering*, Springer, 518–532. Retrieved June 21, 2017 from http://link.springer.com/chapter/10.1007/978-3-642-31095-9_34

[13] Emory S. Bogardus. 1947. Measurement of Personal-Group Relations. *Sociometry* 10, 4 (November 1947), 306. DOI:https://doi.org/10.2307/2785570

[14] David Both. 2020. systemd. In *Using and Administering Linux: Volume 2: Zero to SysAdmin: Advanced Topics*, David Both (ed.). Apress, Berkeley, CA, 379–410. DOI:https://doi.org/10.1007/978-1-4842-5455-4_13

[15] David Both. 2020. D-Bus and udev. In *Using and Administering Linux: Volume 2: Zero to SysAdmin: Advanced Topics*, David Both (ed.). Apress, Berkeley, CA, 411–424. DOI:https://doi.org/10.1007/978-1-4842-5455-4_14

[16] British Standards Institution and British Standards Institution. 1990. *Safety requirements for radio transmitting equipment*. British Standards Institution, London.

[17] César Calderón and Luis Servén. 2004. *The Effects of Infrastructure Development on Growth and Income Distribution*. The World Bank. DOI:https://doi.org/10.1596/1813-9450-3400

[18] Pádraig Carmody. 2012. The Informationalization of Poverty in Africa? Mobile Phones and Economic Structure. *Inf. Technol. Int. Dev.* 8, 3 (September 2012), 1–17.

[19] Roberto Cibin, Sarah Robinson, Kristen M. Scott, Duarte Sousa, Petra Žišt, Laura Maye, Mariacristina Sciannamblo, Simone Ashby, Christopher Csíkszentmihályi, Nadia Pantidi, and Maurizio Teli. 2020. Co-designing convivial tools to support participation in community radio. *Radio JournalInternational Stud. Broadcast Audio Media* 18, 1 (April 2020), 43–61. DOI:https://doi.org/10.1386/rjao_00015_1

[20] Roberto Cibin, Sarah Robinson, Maurizio Teli, Conor Linehan, Laura Maye, and Christopher Csíkszentmihályi. 2020. Shaping Social Innovation in Local Communities: The Contribution of Intermediaries. In *Proceedings of the 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society*, ACM, Tallinn Estonia, 1–12. DOI:https://doi.org/10.1145/3419249.3420178

[21] Roberto Cibin, Maurizio Teli, and Sarah Robinson. 2019. Institutioning and Community Radio. A comparative perspective. In *Proceedings of the 9th International Conference on Communities & Technologies - Transforming Communities*, ACM, Vienna Austria, 143–154. DOI:https://doi.org/10.1145/3328320.3328392

[22] Bev Clark and Brenda Burrell. 2009. Freedom Fone: Dial-up Information Service. In *2009 International Conference on Information and Communication Technologies and Development (ICTD)*, IEEE, Doha, Qatar, 483–483. DOI:https://doi.org/10.1109/ICTD.2009.5426724

[23] Peter da Costa. 2012. The Growing Pains of Community Radio in Africa. *Nord. Rev.* 33, Special Issue (January 2012). DOI:https://doi.org/10.2478/nor-2013-0031

[24] Chris Csíkszentmihályi and Jude Mukundane. 2016. RootIO: ICT+ telephony for grassroots radio. In *IST-Africa Week Conference, 2016*, IEEE, 1–13.

[25] Christopher Csikszentmihalyi, Jude Mukundane, Gemma F. Rodrigues, Daniel Mwesigwa, and Michelle Kasprzak. 2018. The Space of Possibilities: Political Economies of Technology Innovation in Sub-Saharan Africa. ACM Press, 1–13. DOI:https://doi.org/10.1145/3173574.3173880

[26] Christopher Csíkszentmihályi, Gemma Rodrigues, Elsa Ferreira, Cristiano Gianolla, Cátia Jardim, Michelle Kasprzak, Elise Leclerc, Jude Mukundane, Daniel Mwesigwa, Duarte Sousa, and Zanna Rodrigues. SOCIAL TECH ECOSYSTEMS IN SUB-SAHARAN AFRICA. 88.

[27] R. Danymol, T. Ajitha, and R. Gandhiraj. 2013. Real-time communication system design using RTL-SDR and Raspberry Pi. In *2013 International Conference on Advanced Computing and*

*Communication Systems*, IEEE, Coimbatore, India, 1–5. DOI:https://doi.org/10.1109/ICACCS.2013.6938691

[28] Anne Delaporte and Kalvin Bahia. 2021. *The State of Mobile Internet Connectivity 2021*. GSMA.

[29] Martin Dietz, Markus Multrus, Vaclav Eksler, Vladimir Malenovsky, Erik Norvell, Harald Pobloth, Lei Miao, Zhe Wang, Lasse Laaksonen, Adriana Vasilache, Yutaka Kamamoto, Kei Kikuiri, Stephane Ragot, Julien Faure, Hiroyuki Ehara, Vivek Rajendran, Venkatraman Atti, Hosang Sung, Eunmi Oh, Hao Yuan, and Changbao Zhu. 2015. Overview of the EVS codec architecture. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5698–5702. DOI:https://doi.org/10.1109/ICASSP.2015.7179063

[30] Akash Dutt Dubey and Shreya Tripathi. 2020. Analysing the Sentiments towards Work-From-Home Experience during COVID-19 Pandemic. *J. Innov. Manag.* 8, 1 (April 2020). DOI:https://doi.org/10.24840/2183-0606_008.001_0003

[31] Christina Dunbar-Hester. 2016. Frailties at the Borders: Stalled Activist Media Projects in East Africa. (2016), 22.

[32] Kevin C. Dunn. 2004. Uganda: The Lord’s Resistance Army. *Rev. Afr. Polit. Econ.* 31, 99 (2004), 139–142.

[33] Daniel J. Elazar. 1978. The problem of political distance. *Natl. Civ. Rev.* 67, 7 (July 1978), 307–338. DOI:https://doi.org/10.1002/ncr.4100670704

[34] Fadi El-Moussa, Parmindher Mudhar, and Andy Jones. 2009. Overview of SIP attacks and countermeasures. In *International Conference on Information Security and Digital Forensics*, Springer, 82–91.

[35] EN ETSI. 2005. 301 489-11: Electromagnetic compatibility and Radio spectrum Matters (ERM); ElectroMagnetic Compatibility (EMC) standard for radio equipment and services; Part 11: Specific conditions for terrestrial sound broadcasting service transmitters.

[36] EN ETSI. 2005. 302 018-2: Electromagnetic compatibility and Radio spectrum Matters (ERM); Transmitting equipment for the Frequency Modulated (FM) sound broadcasting service; Part 2: Harmonized EN under article 3.2 of the R&TTE Directive.

[37] EN ETSI. 2019. 301 489-1: ElectroMagnetic Compatibility (EMC) standard for radio equipment and services; Part 1: Common technical requirements; Harmonised Standard for ElectroMagnetic Compatibility.

[38] European Commission. Directorate General for Communications Networks, Content and Technology., IHS Markit Ltd., Point Topic., and OMDIA. 2020. *Broadband coverage in Europe 2019: mapping progress towards the coverage objectives of the Digital Agenda : final report.* Publications Office, LU. Retrieved May 19, 2021 from https://data.europa.eu/doi/10.2759/375483

[39] Colin Fraser and Sonia Restrepo Estrada. Community radio handbook; 2005. 105.

[40] Colin Fraser and Sonia Restrepo-Estrada. 2002. Community radio for change and development. *Development* 45, 4 (2002), 69–73.

[41] M Ghosh and Arindam Ghosh. 2014. Financial inclusion strategies of banks: Study of Indian states. *Int. J. Appl. Financ. Manag. Perspect.* 3, 2 (2014), 990.

[42] Alison Gillwald, Onkokame Mothobi, Ali Ndiwalana, and Tusu Tusubira. THE STATE OF ICT IN UGANDA. 36.

[43] Shikoh Gitau, Gary Marsden, and Jonathan Donner. 2010. After access: challenges facing mobile-only internet users in the developing world. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2603–2606. Retrieved September 18, 2017 from http://dl.acm.org/citation.cfm?id=1753720

[44] Dru C. Gladney. 1990. The ethnogenesis of the Uighur. *Cent. Asian Surv.* 9, 1 (January 1990), 1–28. DOI:https://doi.org/10.1080/02634939008400687

[45] GSMA. 2019. *The Mobile Economy 2019*. GSMA Intelligence, London, United Kingdom. Retrieved from https://www.gsmaintelligence.com/research/?file=b9a6e6202ee1d5f787cfebb95d3639c5&download

[46] Daniel Hall. 2015. *Ansible Configuration Management - Second Edition*. Packt Publishing Ltd.

[47] Carl Hartung, Adam Lerer, Yaw Anokwa, Clint Tseng, Waylon Brunette, and Gaetano Borriello. 2010. Open data kit: tools to build information services for developing regions. In *Proceedings of the 4th ACM/IEEE International Conference on Information and Communication Technologies and Development - ICTD '10*, ACM Press, London, United Kingdom, 1–12. DOI:https://doi.org/10.1145/2369220.2369236

[48] Kurtis Heimerl. 2013. Community Cellular Networks. (2013). Retrieved June 9, 2017 from http://escholarship.org/uc/item/5hx504jz.pdf

[49] Yi Huang and Kevin Boyle. Antennas : From Theory to Practice. 379.

[50] Nick Hughes and Susie Lonie. 2007. M-PESA: mobile money for the "unbanked" turning cellphones into 24-hour tellers in Kenya. *Innovations* 2, 1–2 (2007), 63–81.

[51] Institutul National De Statistica. Rezultate | Recensamant 2011. *Recensamantul Populatiei Si Locuintelor*. Retrieved May 19, 2021 from http://www.recensamantromania.ro/rezultate-2/

[52] Anwarul Islam and Keita Tsuji. 2011. Bridging digital divide in Bangladesh: study on community information centers. *Electron. Libr.* 29, 4 (August 2011), 506–522. DOI:https://doi.org/10.1108/02640471111156768

[53] ITU. World Telecommunication/ICT Indicators Database. Retrieved July 10, 2021 from https://www.itu.int/en/ITU-D/Statistics/Documents/DDD/ddd_MOZ.pdf

[54] ITU-T. 1988. G711. Retrieved from https://www.itu.int/rec/T-REC-G.711

[55] ITU-T. 2012. G729. Retrieved from https://www.itu.int/rec/T-REC-G.729-201206-I/en

[56] Steven J. Jackson, Alex Pompe, and Gabriel Krieshok. 2012. Repair worlds: maintenance, repair, and ICT for development in rural Namibia. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work - CSCW '12*, ACM Press, Seattle, Washington, USA, 107. DOI:https://doi.org/10.1145/2145204.2145224

[57] Dhanesh Jain and Mahendra Lalwani. 2017. A Review on Optimal Inclination Angles for Solar Arrays. *Int. J. Renew. Energy Res. IJRER* 7, 3 (September 2017), 1053–1061.

[58] Jeffrey James. 2015. Macroeconomic Consequences of the One Laptop per Child Project: OLPC. *J. Int. Dev.* 27, 1 (January 2015), 144–146. DOI:https://doi.org/10.1002/jid.3023

[59] Esther Han Beol Jang, Philip Garrison, Ronel Vincent Vistal, Maria Theresa D. Cunanan, Maria Theresa Perez, Philip Martinez, Matthew William Johnson, John Andrew Evangelista, Syed Ishtiaque Ahmed, Josephine Dionisio, Mary Claire Aguilar Barela, and Kurtis Heimerl. 2019. Trust and Technology Repair Infrastructures in the Remote Rural Philippines: Navigating Urban-Rural Seams. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW (November 2019), 1–25. DOI:https://doi.org/10.1145/3359201

[60] Stylianos Kasampalis, Pavlos I Lazaridis, Zaharias D Zaharis, Aristotelis Bizopoulos, and John Cosmas. Comparison of Longley-Rice, ITM and ITWOM propagation models for DTV and FM Broadcasting. 7.

[61] Soujanya Katikala. GOOGLE™ PROJECT LOON. 6.

[62] Konstantinos Kazakos, Subhashini Arcot Manikam, Gvs Murthy, Papreen Nahar, Peter Phillimore, Shreyaswi Sathyanath, Pushpendra Singh, Meenu Singh, Pete Wright, Deepika Yadav, Patrick Olivier, Siddhartha Asthana, Madeline Balaam, Mona Duggal, Amey Holden, Limalemla Jamir, Nanda Kishore Kannuri, Saurabh Kumar, and Amarendar Reddy Manindla. 2016. A Real-Time IVR Platform for Community Radio. ACM Press, 343–354. DOI:https://doi.org/10.1145/2858036.2858585

[63] Rakesh Kochhar. 2017. *Middle Class Fortunes in Western Europe*. Luxembourg Income Study. Retrieved from https://www.econstor.eu/bitstream/10419/169262/1/702.pdf

[64] Zahir Koradia, C. Balachandran, Kapil Dadheech, Mayank Shivam, and Aaditeshwar Seth. 2012. Experiences of deploying and commercializing a community radio automation system in India. In *Proceedings of the 2nd ACM Symposium on Computing for Development - ACM DEV '12*, ACM Press, Atlanta, Georgia, 1. DOI:https://doi.org/10.1145/2160601.2160612

[65] Zahir Koradia and Aaditeshwar Seth. 2012. Phonepeti: exploring the role of an answering machine system in a community radio station in india. In *Proceedings of the Fifth International Conference on Information and Communication Technologies and Development*, ACM, 278–288.

[66] Arun Kumar, Sheetal K. Agarwal, and Priyanka Manwani. 2010. The spoken web application framework: user generated content and service creation through low-end mobiles. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A)*, ACM, 2. Retrieved June 8, 2017 from http://dl.acm.org/citation.cfm?id=1805990

[67] Caroline W. Lee. 2010. The Roots of Astroturfing. *Contexts* 9, 1 (February 2010), 73–75. DOI:https://doi.org/10.1525/ctx.2010.9.1.73

[68] J. P. Lindsey. 1989. The Fresnel zone and its interpetive significance. *Lead. Edge* 8, 10 (October 1989), 33–39. DOI:https://doi.org/10.1190/1.1439575

[69] Anita G Longley and Phil L Rice. 1968. *Prediction of tropospheric radio transmission loss over irregular terrain: A computer method-1968*. Institute for Telecommunication Sciences.

[70] Anita G Longley and Phil L Rice. 1968. *Prediction of tropospheric radio transmission loss over irregular terrain: A computer method-1968*. Institute for Telecommunication Sciences.

[71] J Maglicane. 2010. SPLAT! An RF signal propagation, loss and terrain analysis tool. *2010 Accessed August 18 2010* (2010).

[72] Syed S Mahmood, Emily Wroe, Arlan Fuller, and Jennifer Leaning. 2017. The Rohingya people of Myanmar: health, human rights, and identity. *The Lancet* 389, 10081 (May 2017), 1841–1850. DOI:https://doi.org/10.1016/S0140-6736(16)00646-2

[73] Linje Manyozo. 2009. Mobilizing Rural and Community Radio in Africa. *Ecquid Novi Afr. Journal. Stud.* 30, 1 (January 2009), 1–23. DOI:https://doi.org/10.1080/02560054.2009.9653389

[74] Mlondi Mpanza. 2016. MTN monopoly officially over. *Sunday Obesrver*. Retrieved February 16, 2017 from https://www.pressreader.com/

[75] Consider Mudenda, David Johnson, Lisa Parks, and Gertjan van Stam. 2013. Power instability in rural Zambia, case Macha. In *International Conference on e-Infrastructure and e-Services for Developing Countries*, Springer, 260–270.

[76] Nanyang Technological University, Elisa Oreglia, Janaki Srinivasan, and International Institute of Information Technology. 2016. ICT, Intermediaries, and the Transformation of Gendered Power Structures. *MIS Q.* 40, 2 (February 2016), 501–510. DOI:https://doi.org/10.25300/MISQ/2016/40.2.13

[77] David Nemer. 2015. Wired Smartphones: Rethinking the role of community technology centers in the mobile Internet era. (2015), 21.

[78] Dita Nugroho and Michele Lonsdale. 2010. Evaluation of OLPC programs global: a literature review. (2010). Retrieved June 27, 2017 from http://research.acer.edu.au/digital_learning/8/

[79] Francis B Nyamnjoh. 2004. Media ownership and control in Africa in the age of globalization. *Who Owns Media* (2004), 119–34.

[80] Eliezer Ofori Odei-Lartey, Dennis Boateng, Samuel Danso, Anthony Kwarteng, Livesy Abokyi, Seeba Amenga-Etego, Stephaney Gyaase, Kwaku Poku Asante, and Seth Owusu-Agyei. 2016. The application of a biometric identification technique for linking community and hospital data in rural Ghana. *Glob. Health Action* 9, 1 (December 2016), 29854. DOI:https://doi.org/10.3402/gha.v9.29854

[81] OFCOM. 2010. *Prediction of the 'useable' coverage of FM radio services*. Retrieved from https://www.ofcom.org.uk/__data/assets/pdf_file/0013/54310/annex-f.pdf

[82] GIANCARLO PALOMBINI and GEORGETA STOICA. 2014. Melting Traditions and Christmas Carols of a Ukrainian Community (the Hahols) from the Village of Sfantu Gheorghe (Danube Delta). *Mem. Ethnol.* 14, (2014).

[83] Tapan Parikh, Paul Javid, and Kaushik Ghosh. 2006. Mobile Phones and Paper Documents: Evaluating A New Approach for Capturing Microfinance Data in Rural India. (2006), 10.

[84] Neil Patel, Deepti Chittamuru, Anupam Jain, Paresh Dave, and Tapan S. Parikh. 2010. Avaaj otalo: a field study of an interactive voice forum for small farmers in rural india. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 733–742. Retrieved June 7, 2017 from http://dl.acm.org/citation.cfm?id=1753434

[85] Ron Peters (Ed.). 2009. cron. In *Expert Shell Scripting*. Apress, Berkeley, CA, 81–85. DOI:https://doi.org/10.1007/978-1-4302-1842-5_12

[86] Isabella Rega, Lorenzo Cantoni, Sara Vannini, Salomão David, Alexandre Baia, and Gertrudes Macueve. Community Multimedia Centres in Mozambique: a Map. 13.

[87] Everett M Rogers. The Digital Divide. 16.

[88] RootIO. 2016. Low Power FM Transmitter Antenna From Agricultural Tubing. *Instructables*. Retrieved May 20, 2021 from https://www.instructables.com/Low-Power-FM-Transmitter-Antenna-From-Agricultural/

[89] francis rumsey. 2021. broadcasting from home: pandemic and beyond. *J. Audio Eng. Soc.* 68, 12 (December 2021), 987–991.

[90] Nithya Sambasivan, Ed Cutrell, Kentaro Toyama, and Bonnie Nardi. 2010. Intermediated technology use in developing communities. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2583–2592. Retrieved September 18, 2017 from http://dl.acm.org/citation.cfm?id=1753718

[91] Rudy Schusteritsch, Shailendra Rao, and Kerry Rodden. 2005. Mobile Search with Text Messages: Designing the User Experience for Google SMS. (2005), 4.

[92] Yael Schwartzman and Tapan S Parikh. Using CAM-equipped Mobile Phones for Procurement and Quality Control at a Rural Coffee Cooperative. 10.

[93] Kristen M. Scott, Simone Ashby, David A. Braude, and Matthew P. Aylett. 2019. Who owns your voice?: ethically sourced voices for non-commercial tts applications. In *Proceedings of the 1st International Conference on Conversational User Interfaces - CUI '19*, ACM Press, Dublin, Ireland, 1–3. DOI:https://doi.org/10.1145/3342775.3342793

[94] Kristen M. Scott, Simone Ashby, and Roberto Cibin. 2020. Implementing text-to-speech tools for community radio in remote regions of Romania. In *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, ACM, Virtual Event Mexico, 123–126. DOI:https://doi.org/10.1145/3410530.3414410

[95] Kristen M. Scott, Simone Ashby, and Julian Hanna. 2020. "Human, All Too Human": NOAA Weather Radio and the Emotional Impact of Synthetic Voices. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ACM, Honolulu HI USA, 1–9. DOI:https://doi.org/10.1145/3313831.3376338

[96] Kristen M. Scott, Simone Ashby, and Adriana Stan. 2020. Designing a Synthesized Content Feed System for Community Radio. In *Proceedings of the 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society*, ACM, Tallinn Estonia, 1–4. DOI:https://doi.org/10.1145/3419249.3420177

[97] Maung K. Sein and Bjørn Furuholt. 2012. Intermediaries: bridges across the digital divide. *Inf. Technol. Dev.* 18, 4 (October 2012), 332–344. DOI:https://doi.org/10.1080/02681102.2012.667754

[98] Brian Semujju. 2016. Introducing Community Audio Towers as an alternative to community radio in Uganda. *J. Altern. Community Media* 1, 1 (April 2016), 141–153. DOI:https://doi.org/10.1386/joacm_00024_1

[99] Lisa J. Servon and Marla K. Nelson. 2001. Community Technology Centers: Narrowing the Digital Divide in Low-Income, Urban Communities. *J. Urban Aff.* 23, 3–4 (September 2001), 279–290. DOI:https://doi.org/10.1111/0735-2166.00089

[100] Sid Shumate. 2008. Deterministic equations for computer approximation of itu-r p. 1546-2. In *International Symposium on Advanced Radio Technologies/ClimDiff*, 86–95.

[101] Maddalena Sorrentino and Bjoern Niehaves. 2010. Intermediaries in E-Inclusion: A Literature Review. In *2010 43rd Hawaii International Conference on System Sciences*, IEEE, Honolulu, Hawaii, USA, 1–10. DOI:https://doi.org/10.1109/HICSS.2010.239

[102] S. Revi Sterling, John O'Brien, and John K. Bennett. 2007. Advancement through Interactive Radio. In *2007 International Conference on Information and Communication Technologies and Development*, IEEE, Bangalore, India, 1–8. DOI:https://doi.org/10.1109/ICTD.2007.4937393

[103] Daniel M.L. Storisteanu, Toby L. Norman, Alexandra Grigore, and Alain B. Labrique. 2016. Can biometrics beat the developing world's challenges? *Biom. Technol. Today* 2016, 11 (November 2016), 5–9. DOI:https://doi.org/10.1016/S0969-4765(16)30193-X

[104] Ewan Sutherland. 2009. Counting Mobile Phones, Sim Cards & Customers. *SSRN Electron. J.* (2009). DOI:https://doi.org/10.2139/ssrn.1468847

[105] Arame Tall, James Hansen, Alexa Jay, Bruce Campbell, James Kinyangi, Pramod K Aggarwal, and Robert Zougmoré. Scaling up climate services for farmers: Mission Possible. 13 , 44.

[106] UCC, Active Watch, MedAlert, BIPG, RootIO, M-ITI, AMARC. 2020. *Grassroot Wavelengths Project Evaluation Report (D4.1)*. European Commission. Retrieved from https://ec.europa.eu/research/participants/documents/download Public?documentIds=080166e5d7986f55&appId=PPGMS

[107] UNESCO Institute for Statistics (UIS). Retrieved February 7, 2018 from https://data.worldbank.org/indicator/SE.ADT.LITR.ZS?locations=ZF

[108] Eben Upton and Gareth Halfacree. 2012. *Meet the Raspberry Pi*. John Wiley & Sons.

[109] Jakob J Van Zyl. 2001. The Shuttle Radar Topography Mission (SRTM): a breakthrough in remote sensing of topography. *Acta Astronaut.* 48, 5–12 (2001), 559–565.

[110] Aditya Vashistha, Edward Cutrell, Gaetano Borriello, and William Thies. 2015. Sangeet Swara: A Community-Moderated Voice Forum in Rural India. ACM Press, 417–426. DOI:https://doi.org/10.1145/2702123.2702191

[111] Philip Verwimp and Jean-Francois Maystadt. 2015. *Forced Displacement and Refugees in Sub-Saharan Africa: An Economic Inquiry*. The World Bank. DOI:https://doi.org/10.1596/1813-9450-7517

[112] G Vijayaraghavan, Mark Brown, and Malcolm Barnes. 2004. *Practical grounding, bonding, shielding and surge protection*. Newnes, Amsterdam; London. Retrieved May 17, 2021 from http://www.123library.org/book_details/?id=34525

[113] Steve Vinoski. 2006. Advanced Message Queuing Protocol. *IEEE Internet Comput.* 10, 6 (November 2006), 87–89. DOI:https://doi.org/10.1109/MIC.2006.116

[114] Binyavanga Wainaina. 2007. Glory: The Soft Bigotry of Great Expectations. *Bidoun* Technology / Issue 10 (2007), 48–52.

[115] Willems Wendy. 2016. Beyond Free Basics: Facebook, data bundles and Zambia's social media internet. *Africa at LSE*. Retrieved April 24, 2017 from http://blogs.lse.ac.uk/africaatlse/2016/09/01/beyond-free-basics-facebook-data-bundles-and-zambias-social-media-internet/

[116] World Bank. 2021. *Population Estimates and Projections*. Retrieved from https://databank.worldbank.org/source/population-estimates-and-projection

[117] World Bank. 2021. *Access to electricity, rural (% of rural population) - Romania | Data*. Retrieved May 19, 2021 from https://data.worldbank.org/indicator/EG.ELC.ACCS.RU.ZS?locations=RO

[118] World Bank. 2021. *Access to electricity, rural (% of rural population) - Uganda | Data*. Retrieved May 19, 2021 from https://data.worldbank.org/indicator/EG.ELC.ACCS.RU.ZS?locations=UG

[119] Susan P. Wyche and Laura L. Murphy. 2012. "Dead China-make" Phones off the Grid: Investigating and Designing for Mobile Phone Use in Rural Africa. In *Proceedings of the Designing Interactive Systems Conference* (DIS '12), ACM, New York, NY, USA, 186–195. DOI:https://doi.org/10.1145/2317956.2317985

[120] Mark Zuckerberg. 2016. Internet. org: Telecom Infra Project. (2016).

[121] 2013. *Universal Service Fund and Digital Inclusion For All Study*. International Telecommunications Union. Retrieved June 4, 2017 from https://www.itu.int/en/ITU-D/Conferences/GSR/Documents/ITU%20USF%20Final%20Report.pdf

[122] 2017. *State of Electricity Access Report 2017*. World Bank, Washington, D.C. Retrieved February 7, 2018 from http://documents.worldbank.org/curated/en/364571494517675149/pdf/114841-REVISED-JUNE12-FINAL-SEAR-web-REV-optimized.pdf

[123] 2018. Ericsson Mobility Report November 2018. (2018), 32.

[124] 2019. *How Innovation Can Drive Rural Connectivity*. GSMA, London. Retrieved January 21, 2020 from https://www.gsma.com/mobilefordevelopment/wp-content/uploads/2019/07/GSMA-How-Innovation-Can-Drive-Rural-Connectivity-Report-2019.pdf

[125] 2019. Connecting Africa Through Broadband A strategy for doubling connectivity by 2021 and reaching universal access by 2030. Retrieved May 14, 2021 from https://www.broadbandcommission.org/Documents/working-groups/DigitalMoonshotforAfrica_Report.pdf

[126] 2020. Battle in Portugal over plans for compulsory virus app. *France 24*. Retrieved May 15, 2021 from https://www.france24.com/en/live-news/20201016-battle-in-portugal-over-plans-for-compulsory-virus-app

[127] 2021. 5G Spectrum - GSMA Public Policy Position. Retrieved from https://www.gsma.com/spectrum/wp-content/uploads/2021/04/5G-Spectrum-Positions.pdf

[128] 2022. *The Mobile Gender Gap Report 2022*. GSMA, London. Retrieved from https://www.gsma.com/r/wp-content/uploads/2022/06/The-Mobile-Gender-Gap-Report-2022.pdf

[129] Measuring Digital Development: ICT Price Trends 2019. 178.

[130] Starlink. *Starlink*. Retrieved October 21, 2021 from https://www.starlink.com

[131] Airtime Pro: Quick-Start Your Own Internet Radio Station. *Airtime Pro*. Retrieved May 15, 2021 from https://www.airtime.pro/

[132] Coaxial Cable Attenuation Chart. Retrieved May 16, 2021 from http://rfelektronik.se/manuals/Datasheets/Coaxial_Cable_Attenuation_Chart.pdf

[133] Rádios – FORCOM. Retrieved October 21, 2021 from https://www.forcom.org.mz/radios/

[134] The Economics of Rural Radio in Africa: An Introductory Study into the Costs and Revenues. *The Communication Initiative Network*. Retrieved January 19, 2016 from http://www.comminit.com/edutain-africa/content/economics-rural-radio-africa-introductory-study-costs-and-revenues

[135] Uganda Radionetwork. Retrieved May 16, 2021 from https://ugandaradionetwork.net/a/

[136] EUR-Lex - 31999L0005 - EN - EUR-Lex. Retrieved May 17, 2021 from https://eur-lex.europa.eu/legal-content/en/ALL/?uri=CELEX%3A31999L0005

[137] Primaria Comunei Varvoru de Jos. Retrieved May 19, 2021 from http://varvorudejos.ro/prezentare_locala

[138] KissFM - Airplay 100. *KissFM - Asculti doar HITURI!* Retrieved May 20, 2021 from https://www.kissfm.ro/airplay100

[139] Mobile OS market share 2021. *Statista*. Retrieved May 20, 2021 from https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/

[140] RECOMMENDATION ITU-R P.2001-3 - A general purpose wide-range terrestrial propagation model in the frequency range 30 MHz to 50 GHz. 58.

# Annex A: Core Household indicators

| | *Proportion of households with* | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *Radio* | *Year of data* | *TV* | *of data* | *Fixed line telephone* | *Year of data* | *Mobile telephone* | *Year of data* | *Computer* | *Year of data* | *Internet access at home* | *Year of data* |
| Afghanistan | ... | | ... | | ... | | ... | | ... | | ... | |
| Albania | - | 2018 | 99.3 | 2018 | 17.1 | 2018 | 86.6 | 2018 | 20.2 | 2018 | 82.2 | 2019 |
| Algeria | 52.6 | 2018 | 99.0 | 2018 | 27.7 | 2018 | 99.1 | 2018 | 42.2 | 2018 | 74.4 | 2018 |
| Andorra | ... | | ... | | ... | | ... | | 85.7 | 2017 | 94.5 | 2017 |
| Angola | 16.4 | 2018 | 22.9 | 2018 | ... | | ... | | 32.2 | 2018 | 6.7 | 2018 |
| Antigua and Barbuda | ... | | ... | | ... | | ... | | ... | | ... | |
| Argentina | ... | | ... | | ... | | ... | | 64.3 | 2017 | 75.9 | 2017 |
| Armenia | 98.9 | 2016 | 97.0 | 2019 | 33.4 | 2019 | 92.5 | 2019 | 57.9 | 2019 | 76.4 | 2019 |
| Australia | ... | | ... | | ... | | ... | | 82.4 | 2017 | 86.1 | 2017 |
| Austria | ... | | ... | | ... | | ... | | 85.4 | 2017 | 89.9 | 2019 |
| Azerbaijan | 98.9 | 2019 | 100.0 | 2019 | 68.7 | 2019 | 90.6 | 2019 | 65.0 | 2019 | 79.1 | 2019 |
| Bahamas | ... | | ... | | ... | | ... | | ... | | ... | |
| Bahrain | 57.6 | 2018 | 85.9 | 2019 | 7.6 | 2019 | 100.0 | 2019 | 94.8 | 2019 | 99.7 | 2019 |
| Bangladesh | 0.6 | 2019 | 50.6 | 2019 | ... | | 94.8 | 2019 | 5.6 | 2019 | 37.6 | 2019 |
| Barbados | ... | | ... | | ... | | ... | | ... | | ... | |
| Belarus | ... | | 98.2 | 2019 | 91.3 | 2019 | 97.9 | 2019 | 71.8 | 2019 | 78.6 | 2019 |
| Belgium | ... | | 94.6 | 2018 | 63.4 | 2018 | 97.2 | 2018 | 90.1 | 2018 | 89.7 | 2019 |
| Belize | 55.5 | 2019 | 78.1 | 2019 | 6.7 | 2019 | 93.9 | 2019 | 37.2 | 2019 | 60.2 | 2019 |
| Benin | 54.0 | 2017 | 28.7 | 2017 | ... | | 83.9 | 2017 | 4.7 | 2017 | 4.6 | 2017 |
| Bhutan | 23.9 | 2017 | 71.5 | 2017 | 4.0 | 2017 | 97.0 | 2017 | 23.6 | 2017 | 44.3 | 2017 |
| Bolivia (Plurinational State of) | 39.7 | 2019 | 83.9 | 2019 | 13.6 | 2019 | ... | | 27.3 | 2019 | 23.0 | 2019 |
| Bosnia and Herzegovina | 64.9 | 2019 | 98.3 | 2019 | ... | | 90.6 | 2019 | 64.8 | 2019 | 72.0 | 2019 |
| Botswana | ... | | ... | | ... | | ... | | 27.8 | 2019 | 63.5 | 2019 |
| Brazil | 60.8 | 2019 | 95.2 | 2019 | 21.7 | 2019 | 92.9 | 2019 | 39.4 | 2019 | 71.4 | 2019 |
| Brunei Darussalam | 63.0 | 2018 | 91.9 | 2018 | 32.6 | 2018 | 93.4 | 2018 | 64.0 | 2019 | 53.6 | 2019 |
| Bulgaria | ... | | ... | | ... | | ... | | 63.0 | 2017 | 75.1 | 2019 |
| Burkina Faso | 60.4 | 2017 | 25.0 | 2017 | ... | | 93.9 | 2017 | 5.1 | 2017 | ... | |
| Burundi | 36.9 | 2016 | 4.8 | 2016 | ... | | 46.9 | 2016 | 1.0 | 2017 | 0.3 | 2017 |
| Cabo Verde | 47.5 | 2019 | 83.3 | 2019 | 94.3 | 2019 | 91.7 | 2019 | 34.6 | 2019 | 67.0 | 2019 |
| Cambodia | 42.6 | 2017 | 66.6 | 2017 | 1.1 | 2017 | 85.0 | 2017 | 13.3 | 2019 | 41.0 | 2019 |
| Cameroon | 38.8 | 2018 | 51.2 | 2018 | ... | | 85.4 | 2018 | 17.9 | 2018 | 21.7 | 2017 |
| Canada | ... | | ... | | 66.8 | 2016 | 87.9 | 2016 | 84.1 | 2017 | 89.0 | 2017 |
| Central African Rep. | ... | | ... | | ... | | ... | | ... | | ... | |
| Chad | ... | | ... | | ... | | ... | | ... | | ... | |
| Chile | ... | | ... | | ... | | ... | | 60.2 | 2017 | 87.5 | 2017 |
| China | ... | | ... | | ... | | ... | | ... | | ... | |
| Colombia | 69.7 | 2018 | 90.7 | 2018 | 29.2 | 2018 | 95.2 | 2018 | 37.2 | 2019 | 52.2 | 2019 |
| Comoros | ... | | ... | | ... | | ... | | ... | | ... | |
| Congo (Rep. of the) | ... | | ... | | ... | | ... | | ... | | ... | |
| Costa Rica | 69.1 | 2018 | 97.1 | 2018 | 36.6 | 2017 | 96.2 | 2018 | 47.0 | 2019 | 86.3 | 2019 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Côte d'Ivoire | 35.5 | 2019 | 56.7 | 2019 | 1.9 | 2019 | 58.8 | 2019 | 12.0 | 2019 | 16.9 | 2019 |
| Croatia | ... | | ... | | ... | | ... | | 74.3 | 2019 | 80.5 | 2019 |
| Cuba | 52.0 | 2017 | 94.9 | 2017 | ... | | ... | | 17.1 | 2019 | 32.4 | 2019 |
| Cyprus | ... | | ... | | ... | | ... | | 74.8 | 2019 | 89.6 | 2019 |
| Czech Republic | ... | | ... | | ... | | ... | | 77.9 | 2019 | 81.1 | 2019 |
| Dem. People's Rep. of Korea | ... | | ... | | ... | | ... | | ... | | ... | |
| Dem. Rep. of the Congo | 37.6 | 2017 | 19.4 | 2017 | ... | | 46.5[45] | 2017 | 5.3 | 2017 | 1.3 | 2017 |
| Denmark | 28.8 | 2020 | 66.7 | 2020 | 12.9 | 2020 | 81.4 | 2020 | 77.7 | 2020 | 92.5 | 2020 |
| Djibouti | ... | | ... | | ... | | ... | | 36.6 | 2017 | 57.7 | 2017 |
| Dominica | ... | | ... | | ... | | ... | | ... | | ... | |
| Dominican Rep. | 51.8 | 2018 | 84.1 | 2018 | 23.4 | 2018 | 91.6 | 2018 | 27.9 | 2019 | 33.7 | 2019 |
| Ecuador | 25.9 | 2017 | 71.8 | 2017 | 36.9 | 2017 | 90.7 | 2017 | 40.6 | 2019 | 37.2 | 2018 |
| Egypt | 29.7 | 2019 | 98.9 | 2019 | 27.6 | 2019 | 98.8 | 2019 | 64.0 | 2019 | 59.9 | 2019 |
| El Salvador | 29.3 | 2019 | 87.9 | 2019 | 17.9 | 2019 | 94.5 | 2019 | 16.7 | 2019 | 23.5 | 2019 |
| Equatorial Guinea | ... | | ... | | ... | | ... | | ... | | ... | |
| Eritrea | ... | | ... | | ... | | ... | | ... | | ... | |
| Estonia | ... | | ... | | ... | | ... | | 86.9 | 2017 | 90.4 | 2019 |
| Eswatini | ... | | ... | | ... | | ... | | ... | | ... | |
| Ethiopia | ... | | ... | | ... | | ... | | 5.0 | 2016 | 15.4 | 2016 |
| Fiji | ... | | ... | | ... | | ... | | ... | | ... | |
| Finland | ... | | ... | | ... | | ... | | 87.4 | 2019 | 89.0 | 2019 |
| France | ... | | ... | | ... | | ... | | 77.5 | 2017 | 84.0 | 2019 |
| Gabon | ... | | ... | | ... | | ... | | ... | | ... | |
| Gambia | 69.3 | 2018 | 52.6 | 2018 | ... | | 97.3 | 2018 | 18.9 | 2018 | 63.3 | 2018 |
| Georgia | 1.5 | 2018 | 97.2 | 2018 | 30.7 | 2018 | 96.1 | 2018 | 62.0 | 2019 | 79.3 | 2019 |
| Germany | ... | | ... | | ... | | ... | | 88.1 | 2017 | 92.1 | 2020 |
| Ghana | 56.5 | 2019 | 63.7 | 2019 | ... | | 92.3 | 2019 | 15.8 | 2019 | 22.4 | 2017 |
| Greece | ... | | ... | | ... | | ... | | 70.5 | 2017 | 78.5 | 2019 |
| Grenada | ... | | ... | | ... | | ... | | ... | | ... | |
| Guatemala | ... | | ... | | 10.4 | 2019 | 87.1 | 2019 | ... | | 22.7 | 2019 |
| Guinea | 54.4 | 2018 | 31.3 | 2018 | ... | | 89.2 | 2018 | 7.0 | 2018 | 13.0 | 2019 |
| Guinea-Bissau | ... | | ... | | ... | | ... | | ... | | ... | |
| Guyana | ... | | ... | | ... | | ... | | ... | | ... | |
| Haiti | 48.0 | 2016 | 30.7 | 2016 | ... | | 75.9 | 2016 | 11.0 | 2018 | 7.0 | 2018 |
| Honduras | ... | | ... | | ... | | ... | | 17.1 | 2017 | ... | |
| Hong Kong, China | ... | | ... | | ... | | ... | | 77.6 | 2019 | 94.1 | 2019 |
| Hungary | ... | | ... | | ... | | ... | | ... | | 86.2 | 2019 |
| Iceland | ... | | ... | | ... | | ... | | ... | | ... | |
| India | ... | | ... | | ... | | ... | | 10.7 | 2018 | 23.8 | 2018 |
| Indonesia | 23.0 | 2017 | 14.2 | 2018 | 2.1 | 2019 | 90.2 | 2019 | 18.8 | 2019 | 73.7 | 2019 |
| Iran (Islamic Republic of) | 92.7 | 2017 | 99.6 | 2017 | 78.3 | 2017 | 95.8 | 2017 | 74.0 | 2018 | 79.6 | 2018 |
| Iraq | 7.9 | 2018 | 99.6 | 2019 | 0.7 | 2019 | 99.2 | 2019 | 81.2 | 2018 | 73.1 | 2018 |
| Ireland | ... | | ... | | ... | | ... | | 85.7 | 2018 | 89.1 | 2018 |
| Israel | ... | | 87.8 | 2018 | 56.0 | 2018 | 97.9 | 2018 | 79.5 | 2019 | 75.9 | 2019 |
| Italy | ... | | 96.6 | 2017 | 54.5 | 2017 | 95.0 | 2017 | 66.2 | 2019 | 76.1 | 2019 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Jamaica | 73.3 | 2017 | 89.7 | 2017 | 11.6 | 2017 | 95.1 | 2017 | 36.4 | 2017 | 52.9 | 2017 |
| Japan | ... | | ... | | 69.9 | 2019 | 97.0 | 2019 | 74.6 | 2019 | 96.9 | 2019 |
| Jordan | 10.0 | 2017 | 98.8 | 2017 | ... | | 97.6 | 2017 | 42.9 | 2017 | 37.4 | 2017 |
| Kazakhstan | ... | | 98.6 | 2019 | 56.2 | 2019 | 98.6 | 2019 | 80.5 | 2019 | 90.3 | 2019 |
| Kenya | ... | | ... | | ... | | ... | | 8.8 | 2019 | 17.9 | 2019 |
| Kiribati | 44.3 | 2018 | 1.4 | 2018 | ... | | 68.3 | 2018 | 31.4 | 2018 | 47.2 | 2018 |
| Korea (Rep. of) | ... | | 98.2 | 2019 | 27.6 | 2019 | 99.8 | 2019 | 71.7 | 2019 | 99.7 | 2019 |
| Kuwait | 71.6 | 2018 | 99.3 | 2018 | 53.0 | 2018 | 99.3 | 2018 | 84.0 | 2019 | 100.0 | 2019 |
| Kyrgyzstan | 16.2 | 2018 | 98.6 | 2019 | 17.5 | 2019 | 92.5 | 2019 | 12.4 | 2019 | 70.1 | 2018 |
| Lao P.D.R. | 20.6 | 2017 | 79.3 | 2017 | ... | | 90.1 | 2017 | 13.5 | 2017 | 1.7 | 2017 |
| Latvia | ... | | ... | | ... | | ... | | 77.4 | 2017 | 89.7 | 2020 |
| Lebanon | ... | | ... | | ... | | ... | | ... | | ... | |
| Lesotho | 51.5 | 2018 | 32.0 | 2018 | 1.9 | 2016 | 90.2 | 2018 | 13.4 | 2018 | 36.9 | 2018 |
| Liberia | 52.1 | 2016 | 22.0 | 2016 | ... | | 62.6 | 2016 | 6.2 | 2016 | ... | |
| Libya | ... | | ... | | ... | | ... | | ... | | ... | |
| Liechtenstein | ... | | ... | | ... | | ... | | ... | | ... | |
| Lithuania | ... | | ... | | ... | | ... | | 76.7 | 2019 | 81.5 | 2019 |
| Luxembourg | ... | | ... | | ... | | ... | | ... | | 93.0 | 2018 |
| Macao, China | ... | | ... | | ... | | ... | | 75.6 | 2019 | 92.3 | 2019 |
| Madagascar | 49.3 | 2018 | 19.8 | 2018 | ... | | 49.6 | 2018 | 5.2 | 2018 | 13.3 | 2018 |
| Malawi | 33.6 | 2018 | 11.8 | 2018 | 1.5 | 2018 | 51.7 | 2018 | 4.2 | 2018 | 10.5 | 2018 |
| Malaysia | 97.2 | 2019 | 97.6 | 2019 | 23.5 | 2019 | 98.2 | 2019 | 71.3 | 2019 | 90.1 | 2019 |
| Maldives | 58.5 | 2016 | 93.8 | 2016 | ... | | 98.7 | 2016 | 70.1 | 2016 | 67.2 | 2016 |
| Mali | 66.4 | 2019 | 42.5 | 2019 | 1.1 | 2019 | 83.9 | 2019 | 4.6 | 2019 | 22.2 | 2019 |
| Malta | ... | | ... | | ... | | ... | | 82.9[46] | 2017 | 86.1 | 2019 |
| Marshall Islands | ... | | ... | | ... | | ... | | ... | | ... | |
| Mauritania | ... | | ... | | ... | | ... | | ... | | ... | |
| Mauritius | ... | | 97.7 | 2018 | 70.5 | 2018 | 94.7 | 2018 | 51.0 | 2019 | 72.9 | 2019 |
| Mexico | 53.9 | 2019 | 92.5 | 2019 | 39.5 | 2019 | 89.4 | 2019 | 44.3 | 2019 | 56.4 | 2019 |
| Micronesia | ... | | ... | | ... | | ... | | ... | | ... | |
| Moldova | ... | | 97.1 | 2019 | 79.2 | 2019 | 88.7 | 2019 | 59.5 | 2019 | 60.8 | 2019 |
| Monaco | ... | | ... | | ... | | ... | | ... | | ... | |
| Mongolia | 6.2 | 2018 | 97.0 | 2019 | 4.4 | 2019 | 97.7 | 2019 | 29.9 | 2019 | 46.7 | 2019 |
| Montenegro | 52.3 | 2018 | 99.1 | 2018 | ... | | 98.4 | 2018 | 61.1 | 2018 | 74.3 | 2019 |
| Morocco | ... | | ... | | 21.8 | 2018 | 99.9 | 2018 | 60.4 | 2019 | 80.8 | 2019 |
| Mozambique | 35.6 | 2018 | 25.5 | 2018 | 1.9 | 2017 | 62.8 | 2018 | 6.7 | 2018 | 2.2 | 2017 |
| Myanmar | 15.2 | 2017 | 54.5 | 2017 | ... | | 81.5 | 2017 | 3.4[48] | 2017 | ... | |
| Namibia | ... | | ... | | ... | | ... | | ... | | ... | |
| Nauru | ... | | ... | | ... | | ... | | ... | | ... | |
| Nepal (Republic of) | 29.3 | 2016 | 51.6 | 2016 | ... | | 92.8 | 2016 | 12.7 | 2016 | ... | |
| Netherlands | ... | | ... | | ... | | ... | | 91.1 | 2019 | 96.2 | 2019 |
| New Zealand | ... | | ... | | ... | | ... | | ... | | ... | |
| Nicaragua | ... | | ... | | ... | | ... | | ... | | ... | |
| Niger | 37.0 | 2018 | 14.1 | 2018 | 0.4 | 2018 | 59.8 | 2018 | 12.0 | 2018 | 9.8 | 2018 |
| Nigeria | 60.6 | 2018 | 49.1 | 2018 | 0.9 | 2017 | 87.9 | 2018 | 6.4 | 2018 | 7.5 | 2017 |
| North Macedonia | 30.8 | 2018 | 98.7 | 2018 | ... | | 97.0 | 2018 | 69.5 | 2018 | 79.3 | 2018 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Norway | ... | | ... | | ... | | ... | | 95.0 | 2017 | 96.1 | 2020 |
| Oman | ... | | ... | | ... | | 97.0 | 2019 | 96.1 | 2019 | 94.4 | 2019 |
| Pakistan | 6.4 | 2017 | 62.8 | 2017 | 4.9 | 2016 | 93.9 | 2017 | 14.3 | 2019 | 34.1 | 2019 |
| Palestine | 55.6 | 2019 | 90.7 | 2019 | 31.2 | 2019 | 97.3 | 2019 | 33.2 | 2019 | 79.6 | 2019 |
| Panama | ... | | ... | | ... | | ... | | 37.3 | 2019 | 70.7 | 2019 |
| Papua New Guinea | 23.8 | 2016 | 12.6 | 2016 | ... | | 56.4 | 2016 | 10.7 | 2016 | ... | |
| Paraguay | 73.7 | 2019 | 90.0 | 2019 | 8.2 | 2019 | 96.7 | 2019 | 26.2 | 2019 | 28.3 | 2019 |
| Peru | 72.6 | 2019 | 80.7 | 2019 | 19.1 | 2019 | 92.1 | 2019 | 32.1 | 2019 | 35.9 | 2019 |
| Philippines | 51.2 | 2017 | 76.9 | 2017 | 6.7[49] | 2016 | 88.8 | 2017 | 23.8 | 2019 | 17.7 | 2019 |
| Poland | ... | | ... | | ... | | ... | | 83.1 | 2019 | 86.7 | 2019 |
| Portugal | ... | | ... | | ... | | ... | | 71.5 | 2017 | 81.7 | 2019 |
| Qatar | 12.9 | 2019 | 96.5 | 2019 | 80.7 | 2019 | 100.0 | 2019 | 87.4 | 2019 | 93.6 | 2019 |
| Romania | ... | | ... | | ... | | ... | | 73.0 | 2017 | 83.6 | 2019 |
| Russian Federation | ... | | ... | | ... | | ... | | 69.4 | 2019 | 76.9 | 2019 |
| Rwanda | 42.0 | 2017 | 10.7 | 2017 | ... | | 62.6 | 2017 | 2.5 | 2017 | 9.3 | 2017 |
| Saint Kitts and Nevis | ... | | ... | | ... | | ... | | ... | | ... | |
| Saint Lucia | ... | | ... | | ... | | ... | | ... | | ... | |
| Saint Vincent and the Grenadines | ... | | ... | | ... | | ... | | 32.5 | 2017 | 54.9 | 2017 |
| Samoa | 61.9 | 2016 | 81.7 | 2016 | 6.4 | 2016 | 90.0 | 2016 | 30.6 | 2016 | 3.8 | 2016 |
| San Marino | ... | | ... | | ... | | ... | | ... | | ... | |
| Sao Tome and Principe | ... | | ... | | ... | | ... | | ... | | ... | |
| Saudi Arabia | ... | | ... | | 14.8 | 2019 | 99.7 | 2019 | 86.2 | 2019 | 99.2 | 2019 |
| Senegal | 65.7 | 2018 | 60.6 | 2018 | ... | | 96.9 | 2018 | 15.8 | 2018 | 6.6 | 2018 |
| Serbia | 69.6 | 2018 | 99.1 | 2018 | ... | | ... | | 73.1 | 2019 | 80.1 | 2019 |
| Seychelles | ... | | ... | | ... | | ... | | ... | | ... | |
| Sierra Leone | 54.7 | 2017 | 18.2 | 2017 | ... | | 65.1 | 2017 | 5.7 | 2017 | 13.8 | 2017 |
| Singapore | 63.0 | 2017 | 94.9 | 2017 | 82.6 | 2017 | 98.2 | 2017 | 88.8 | 2019 | 98.4 | 2019 |
| Slovakia | ... | | ... | | ... | | ... | | 78.9 | 2017 | 82.2 | 2019 |
| Slovenia | ... | | ... | | ... | | ... | | 79.5[46] | 2017 | 89.0 | 2019 |
| Solomon Islands | ... | | ... | | ... | | ... | | ... | | ... | |
| Somalia | ... | | ... | | ... | | ... | | ... | | ... | |
| South Africa | 51.3 | 2017 | 81.5 | 2017 | 8.3 | 2017 | 95.3 | 2017 | 21.9 | 2017 | 61.8 | 2017 |
| South Sudan | ... | | ... | | ... | | ... | | ... | | ... | |
| Spain | 67.8 | 2019 | 99.1 | 2019 | 74.9 | 2019 | 98.5 | 2019 | 80.9 | 2019 | 91.4 | 2019 |
| Sri Lanka | ... | | ... | | ... | | ... | | ... | | ... | |
| Sudan | ... | | ... | | ... | | ... | | ... | | ... | |
| Suriname | 70.6 | 2018 | 87.7 | 2018 | 39.4 | 2019 | 94.4 | 2019 | 37.8 | 2019 | 33.3 | 2019 |
| Sweden | ... | | ... | | ... | | ... | | ... | | 96.1 | 2019 |
| Switzerland | ... | | ... | | ... | | ... | | ... | | 91.6 | 2019 |
| Syrian Arab Republic | ... | | ... | | ... | | ... | | ... | | ... | |
| Tajikistan | 18.1 | 2017 | 98.0 | 2017 | ... | | 96.3 | 2017 | 21.1 | 2017 | 26.7 | 2017 |
| Tanzania | 49.1 | 2017 | 23.0 | 2017 | ... | | 81.5 | 2017 | 3.1 | 2017 | ... | |
| Thailand | ... | | ... | | 7.9 | 2019 | 95.9 | 2018 | 15.9 | 2019 | 74.6 | 2019 |
| Timor-Leste | 24.5 | 2016 | 40.2 | 2016 | ... | | 84.3 | 2016 | 10.9 | 2016 | ... | |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Togo | 56.0 | 2017 | 36.5 | 2017 | ... | | 84.4 | 2017 | 9.5 | 2017 | 26.5 | 2017 |
| Tonga | ... | | ... | | ... | | ... | | ... | | ... | |
| Trinidad and Tobago | ... | | ... | | ... | | ... | | ... | | ... | |
| Tunisia | 46.3 | 2018 | 97.3 | 2018 | ... | | 95.6 | 2018 | 52.1 | 2019 | 51.5 | 2019 |
| Turkey | ... | | ... | | 20.2 | 2017 | 99.4 | 2020 | 52.1 | 2020 | 90.7 | 2020 |
| Turkmenistan | 19.0 | 2019 | 99.8 | 2019 | ... | | 99.1 | 2019 | 43.8 | 2019 | 48.9 | 2019 |
| Tuvalu | ... | | ... | | ... | | ... | | ... | | ... | |
| Uganda | 59.0[51] | 2018 | 19.1[51] | 2018 | ... | | 76.9[51] | 2018 | 3.5[51] | 2018 | ... | |
| Ukraine | 1.0 | 2018 | 94.7 | 2018 | 19.3 | 2018 | 97.5 | 2018 | 62.3 | 2018 | 61.9 | 2018 |
| United Arab Emirates | 18.6 | 2019 | 96.2 | 2019 | 84.7 | 2019 | 100.0 | 2019 | 96.1 | 2019 | 99.1 | 2019 |
| United Kingdom | ... | | ... | | ... | | ... | | 87.5 | 2017 | 95.2 | 2020 |
| United States | ... | | ... | | ... | | ... | | 83.1 | 2019 | 86.6 | 2019 |
| Uruguay | 76.3 | 2019 | 96.2 | 2019 | 62.8 | 2018 | 94.8 | 2018 | 68.2 | 2019 | 69.3 | 2019 |
| Uzbekistan | 20.8 | 2018 | 100.0 | 2018 | 20.4 | 2018 | 99.4 | 2018 | 41.2 | 2019 | 80.4 | 2018 |
| Vanuatu | ... | | ... | | ... | | ... | | ... | | ... | |
| Vatican | ... | | ... | | ... | | ... | | ... | | ... | |
| Venezuela | ... | | ... | | ... | | ... | | ... | | ... | |
| Viet Nam | ... | | ... | | ... | | ... | | 25.8 | 2019 | 46.0 | 2018 |
| Yemen | ... | | ... | | ... | | ... | | ... | | ... | |
| Zambia | 47.1 | 2018 | 36.6 | 2018 | ... | | 73.7 | 2018 | 8.1 | 2018 | 17.7 | 2018 |
| Zimbabwe | 40.2 | 2018 | 35.7 | 2018 | ... | | 87.6 | 2018 | 15.2 | 2018 | 30.3 | 2018 |

# Annex B: Licensing requirements in contexts where RootIO was deployed

UGANDA COMMUNICATIONS COMMISSION

**Application Form UCC-BD/CSP/13/002**

## APPLICATION FOR A BROADCASTING SERVICE LICENSE

**The Form below is for general information of the applicant.**
*(Sections 1 and 2 to be completed in blue or black ink and in Block letters)*

| SECTION 1: CONTACT INFORMATION | |
|---|---|
| **Names:** | |
| **Public Contact Address:** | |
| **Telephone:** Office<br>Cell phone/Mobile<br>**E-mail:**<br>**Fax:** | |

| SECTION 2: BROADCAST LICENCE CATEGORY<br>*(Tick the appropriate Box for the type of service being applied for)* | |
|---|---|
| Public broadcasting services | ☐ |
| Commercial Broadcasting Services | ☐ |
| Community broadcasting Services | |
| Internet Protocol television (IPTV) Service | ☐ |
| Cable Television subscription Service | ☐ |
| Terrestrial subscription broadcasting Service | ☐ |
| Satellite subscription television broadcasting Service | ☐ |
| Broadcasting Subscription Management Service | ☐ |
| Digital Mobile television service | ☐ |

Landing Rights (satellite broadcasting, cable, etc.)

Other (please specify)

*Grassroots Low power networked FM radio transmission. (Aspects of this differ significantly from community broadcast as will be detailed.*

**SECTION 2A: LICENCE CATEGORY**

*(Tick the appropriate Box for the license category being applied for)*

Public ☐

Commercial Subscription ☐ Free to Air ☐

Non-commercial ☐

Community ☐

Broadcasting Subscription Management Service ☐

Landing Rights (satellite broadcasting, cable, etc.) ☐

**SECTION 2B: BROADCASTING SERVICE CATEGORY**

*(Tick the appropriate Box for the broadcasting service being applied for)*

**Radio** ☐ **Television** ☐

**SECTION 2C: PLATFORM CATEGORY**

*(Tick the appropriate Box for the type of platform being applied for)*

Terrestrial ☐

Internet Protocol ☐

Cable ☐

Satellite ☐

Digital Mobile ☐

**SECTION 3: REQUIREMENTS FOR AWARD OF A CONTENT SERVICE PROVIDER'S LICENCE**

**3.1 CORPORATE STATUS** (*Provide original /certified photocopies of the documents)*

a) A covering letter, signed by the applicant, addressed to the Executive Director
b) Certified Copy of Certificate of Incorporation.
c) Memorandum and Articles of Association.
d) Identification or passports of all Directors and Shareholders, as applicable.
e) NGO constitution, Cooperative society Bylaws, Partnership deed or society constitution, as applicable.
f) Valid Tax Compliance, Tax clearance, TIN Number

**3.2 BUSINESS PLAN** *(To be provided in the form of a business Plan)*

### 3.2.1 Market Information

a) Provide the population profile for the area the license is being applied for.
b) Provide market study of the area the license is being sought.
c) Provide information on other broadcasting services in those areas/locality.
d) For subscription services, provide the projected subscriber base for each subscribed broadcasting category.
e) For community services, provide the evidence of local demand or support for the provision of a proposed community broadcasting service in that community.

### 3.2.2 Station Management

a) Provide an organization chart explaining duties and responsibilities of the team managing the broadcasting station.
b) Provide names, address, telephone numbers and attach CVs of key station management staff who shall include among others include Chief executive officer; Finance/Credit Controller, Program coordinator/Manager; Technical Manager, Customer Manager etc.
c) For community broadcasting services, please provide information on how members of the target community will contribute to the sustenance of the community radio station and also participate in the operation and management of the service.

### 3.2.3 Station Financing

a) Provide cash flow projections showing realistic expected levels of costs and revenue over a period of the license term in particular with regard to:
   - Costs for capital investment (equipment, building, offices, vehicles etc.);
   - Recurring costs (staff salaries, royalties, rent energy communications etc.);
   - Financial investments from the applicant(s).
b) Anticipated revenue from advertisements and sponsorships.
c) Are there any commitments from advertisers or sponsors? If so, show proof.
d) Give an accurate estimation of the jobs you anticipate creating from the broadcasting 4service.
   - Full time.
   - Part time.
   - Freelance
e) For community radio, please provide evidence of startup capital and how the community station will be sustained
f) For subscription services, please,
   i) Attach a financial guarantee equivalent from a reputable financial institution
   ii) Provide information on subscription fees, installation fees, relocation fees, maintenance fees, deposits, receiving equipment rental or purchase fees, Subscription fees (for basic, premium packages)and how frequent the fee shall be levied (monthly, etc.)

## SECTION 4: SPECIFIC FORMS TO BE FILLED FOR LICENSE CATEGORY

Attached **Form UCC-BD/CSP/13/002-1**: Content Programming
Attached **Form UCC-BD/CSP/13/002-2**: Technical Information
Attached **Form UCC-BD/CSP/13/002-3:** Subscription services

Applicants, except where the Commission has provided otherwise in a condition of license, shall comply with the regulatory requirements set out in the *Regulations*, and as shall be amended from time to time.

I hereby declare that the information given in this application and any additional documentation associated with this application, to the best of my knowledge and belief is correct.

……………………………………………………………………………….*(Name of person)*

…………………………………………………………………………………*(Title or person among the applicant group)*

………………………………………………………………………………..*(Date)*
If the requested information is the electronic format, it can be sent to: ucc@ucc.co.ug
If the information is in hard copy, please send to hand deliver to the following address:

Executive Director
Uganda communications Commission
UCC House
Plot 42-44 Spring Road, Bugolobi
P.O. Box 7376,
**Kampala, UGANDA**

## APPLICATION FOR CONTENT SERVICE PROVISION

### SECTION A: PROPOSED SERVICE

**Please indicate which genres will be included in your service.**

i. Arts ☐

ii. Chat ☐

iii. Children 's ☐

iv. Content aimed at a particular ethnic group ☐

v. Education ☐

vi. Factual ☐

vii. Films ☐

viii. Gaming ☐

ix. General entertainment ☐

x. Leisure ☐

xi. Local ☐

xii. Music ☐

xiii. News ☐

xiv. Other (please indicate)

a. Quiz ☐

b. Religious ☐

c. Sport ☐

d. Teleshopping ☐

e. User generated content ☐

### SECTION B: PROGRAMMING

*(Where appropriate the information should be provided as an attachment to this application)*

a) Indicate your proposed hours of broadcasting for each day of the week.

b) Show how your proposed programming format is unique from the rest and explain how it will add value to the diversity of broadcasting services in the proposed coverage area.

c) Provide a schedule of program segment for each day of the week

| | |
|---|---|
| d) | Show how your proposed programming format is unique from the rest and explain how it will add value to the diversity of broadcasting services in the proposed coverage area |
| e) | Provide own produced programs and their weekly percentage proportion |
| f) | Provide any planned external sources of your programming (local or foreign). |
| g) | Indicate if the proposed broadcasting service will provide for regular news services and programs on matters of public interest and if so, how often |
| h) | Provide the principal sources of news and information you plan for your programming in approximate percentage terms |
| i) | List domestic sources, which you intend to use for news and information programming |
| j) | List the foreign news sources you plan to use for news and information programming |
| | |
| **SECTION C: DEMAND, NEED AND SUPPORT FOR PROPOSED SERVICE** *(Where appropriate the information should be provided as an attachment to this application)* | |
| a. | To what extent, and in what way, is the proposed program service designed to cater for the tastes and interests of the persons living in the area', either general or particular? |
| b. | If the program service is designed to appeal especially to particular sections of demographic groups (e.g., certain age groups) within the population, state which. |
| c. | If the program service is designed to appeal especially to particular sections of demographic groups (e.g., certain age groups) within the population, state which. |
| d. | To what extent, and in what way, will the proposed service cater for tastes and interests different from those catered for by any other existing broadcasting service within part or all the license area? |
| e. | How will your programming bring social or economic benefits to an area, or to different categories of persons living in the area? |
| f. | How will your programming cater for the tastes, interests and needs of people in the area? |
| g. | How will your programming facilitate civic understanding and fair and well-informed debate through coverage of local news and current affairs? |
| h. | How will your programming facilitate civic understanding and fair and well-informed debate through coverage of local news and current affairs? |
| i. | How will your programming reflect the lives and concerns of communities and cultural interest and traditions in the area? |
| j. | How will your programming include content that informs, educates and entertains, and is not otherwise available through a digital service which is available across Uganda? |
| | |
| **SECTION D: COMMUNITY BROADCASTERS** *(Where appropriate the information should be provided as an attachment to this application)* | |
| a) | Describe in as much detail as possible the community that you propose to serve detailing the current and future needs of that community (provide evidence in support of your claims). |
| b) | What are the objects or aims of the organization/community? |
| c) | How will the applicant continue to identify and serve the future needs of its community? |
| d) | How does the applicant intend to gather information about the needs of the community? |

| | |
|---|---|
| e) | If the organization/community is membership based, provide a copy of the applicant's membership application form and list of members including their addresses. |
| f) | How many of these are financiers? (Please advise if you wish the names and addresses of members to be kept confidential). |
| g) | What is the current membership fee (if any)? |
| h) | If the organization is not membership based, how will members of the community to be served participate in the operations of the service? |
| i) | List the names of office bearers, if any, since registration |
| j) | How does the applicant appoint office bearers? |
| k) | Who will be responsible for the day to day running of your organization? |
| l) | What authority will this entity have for making decisions? |
| *m)* | How will this entity be appointed and to whom will it report? |
| n) | What checks and balances are in place to ensure the service remains a community asset? |
| o) | What entity will be responsible for the control of the station and how will the community access and participation be realized? |
| p) | What measures will be put in place to ensure that the service will be operated on a non-profit basis? |
| *q)* | Provide a business plan that you have prepared to help you meet your organizational/community objectives. |
| r) | Provide details of each existing media service (radio and television) that also serves the community needs that you propose to serve. |
| s) | To what extent do these existing media services meet the needs of the community the applicant proposes to serve |
| t) | If the applicant was allocated a community television/radio license, would it be willing to provide access to unsuccessful applicants? If so, how would the applicant achieve this? |
| *u)* | What steps, if any, has the applicant taken to merge, negotiate some type of co-operative arrangement, with any group that represents a similar community interest in the same license area? |
| *v)* | What, if any, measures will be taken to encourage community access and participation in the day-to-day operations of the proposed service and the selection, provision, and development of programs. |

## TECHNICAL ASSESEMENT FORMAT FOR BROADCASTING CONTENT SERVICE PROVISION

*(The applicant should submit the required information in the format described below)*

| NO. | SCOPE | REQUIREMENTS | DETAILS |
|---|---|---|---|
| **1** | License Details | 1. License Type: Permanent/ Temporary<br>2. Service type: Radio/ Television<br>3. Service Category: Encrypted / FTA<br>4. The names of areas that the licensee wishes to provide the broadcasting service<br>5. Proposed call sign to be used<br>6. If any, the call sign of other licensed services provided by the same company on the same broadcast platform | *The information is to be used for application evaluation and allocation of the appropriate additional resources assignable to the applicant such as logical channel numbers.* |
| **2** | Administrative Information | 1) Company name, trade name, postal address and physical address in Uganda, email address, fax details and telephone numbers.<br>2) Legal Representative: Name of individual/ firm, Telephone contact and email address.<br>3) Technical Representative: Name of individual or firm, Telephone contact and email address.<br>4) Physical location of your administrative offices. This should include the street name, plot number and district. | *The Legal and Technical representatives will be the contact persons that the Commission will liaise with in case of legal and technical issues respectively that need to be addressed by the licensee.* |
| **3.** | Production and/ Aggregation facilities | 1. Physical location of the production/ aggregation facility<br>2. Confirmation of the following operational requirements at the production and / aggregation facility where applicable:<br>• *Access Control* | *The information will be used for the following purposes:-*<br>• *Assessment of audio and video quality of the program streams.* |

| | | | |
|---|---|---|---|
| | | • *Fire Detection and Prevention equipment*<br>• *Cabling and Trunking*<br>• *Electrical and Mechanical Installations for RF transmitters, & Air Conditioning*<br>• *Electrical Clearance (Provide electrical installation certificate)*<br>• *Lightning protection equipment*<br>• *Design of the studio rooms with standard acoustic treatment materials designed and constructed by a registered professional company. The name of the company should be provided*.<br>3. Technical specifications of the following equipment:<br>• *Production and/aggregation equipment,*<br>• *Studio equipment and accessories,*<br>• *Quality monitoring equipment,*<br>• *Power backup equipment*<br>4. Diagram of the studio layout | • *Facilitate identification of possible points of failures in the content service provider network*. |
| **4.** | Maintenance schedule | 1. Contact details of the Technical Representative:<br>• *Name of individual or firm*<br>• *Telephone contact and email address*<br>2. The Maintenance schedule should include:<br>• *Descriptions of the projected software and hardware maintenance and upgrades.*<br>• *Projected number of times that maintenance activities and system upgrades will be carried out.*<br>• *Projected outages as a result of the maintenance and upgrades (if any).* | *Technical representative is the individual /firm that is in charge of the maintenance activities of the broadcasting facilities.*<br><br>*The maintenance schedule framework is required:-*<br>✓ *To assess the ability of the company to guarantee service availability*<br>✓ *To know the scheduled maintenance activities and their effect to provision of services to the consumers if any.* |

| | | | |
|---|---|---|---|
| | | • *Any projected effects on any of your customer receiver equipment (if any).* | |
| **5.** | Business Continuity framework | 1. Disaster recovery plan<br>2. Remote backup plan<br>3. Redundancy Plan<br>4. Electro-mechanical safety measures.<br>5. Equipment and program insurance<br>6. Power backup plan | • *This framework will show how the company can guarantee service availability in the face of human disasters and any operational calamities. Some of these are but not limited to: Fires, vandalism, and virus attacks.*<br>• *The purpose of the remote backup is that it mirrors the important data like subscriber accounts, and programming details off site in case of physical damage, and as such the company always has a reference point.*<br>• *The redundancy plan shows how the Company can seamlessly continue offering service in the event of a system component breakdown.*<br>• *Power backup plan, insurance of programs and equipment ensure service availability.* |
| **6.** | Studio outputs and connection options to transmission broadcasting platforms | 1) Diagram of the system configuration indicating the connection between the studio and transmission platform of the licensed infrastructure providers, connection between external studio (external content) and the local studio.<br>2) Description of the following items:-<br>• *Production format: Analogue / digital*<br>• *Production standard: SD/ HD*<br>• *Studio output format: Analogue/ digital (SDI)*<br>• *Connection interfaces for the projected broadcasting platforms.* | The information will be help the Commission to know:-<br>✓ *How the content services providers will be uploading the program channels onto transmission platforms.*<br>✓ *The production format and standard, so as to facilitate subjective and objective quality of service assessment exercise.*<br>✓ *The studio output that will used to address the compatibility issues that may arise from the connection interfaces with the transmission broadcasting platforms.* |

# Annex C: Questionnaire used to assess community ICT readiness in Northern Uganda

Location:

Age:

Gender:

**Lifestyle and social**

Education Level:

Can you read and write:
a. In your native language?
b. In the official language of your country?

What is your occupation:

If you are employed/studying, do you use a computer at your place of work/school?

How much do you earn per month or per day?

On what do you mostly spend your money?

What is your ethnicity/tribe?

What languages do you speak (in order of fluency)? How frequently do you use each of them? *(Not important, high chance of uniform answer)*

How often do you travel: (this can be adapted for European contexts)

1. To the district center:...................................... Transport Means:...................................
2. To a regional trading hub:...................................... Transport Means:..............................
3. To the capital city
4. Out of the country

Which energy sources do you use for:
a. Cooking
b. Charging personal phone
c. Accessing entertainment
d. Lighting

**Media use and consumption**

What is your main source of:
a. Entertainment?
b. Mass information?
c. News?
d. Announcements and events?

Which of these do you have access to? :

- Television
- FM radio
- Computer
- Basic feature phone

- Smartphone
- Landline phone
- Tablet

Which of these do you own?

- Television
- FM radio
- Computer
- Basic feature phone
- Smartphone
- Landline phone
- Tablet

Which FM radio stations do you listen to mostly?

Have you ever called into a radio station to talk live on air?

Have you ever used the radio station to announce an event, advertise a product/service or procure a product/service?

What are you doing while listening to the radio?

Which programs do you enjoy mostly on the radio?

Which television stations do you watch mostly?

Which are your favorite shows on television?

Which telecommunication network are you subscribed to and why?

Which services do you use from your telecommunications service provider?

a. Voice telephony
b. SMS
c. Mobile Internet
d. Fixed Internet
e. Cable Television (not relevant in ug)

If you use the Internet, what do you use it mostly for/which platforms?

a. Facebook
b. WhatsApp
c. Twitter
d. General browsing
e. Other:..............................................................

If you do not use the Internet, what is the biggest challenge to using the Internet? (include categories)

How much do you spend on: *(this can be replaced by what plan is being subscribed to)*

a. Voice Telephony?

b. SMS texting?

c. Mobile Internet?

Which mobile bundles do you use?

1. Bundle:......................... Cost:......................................
2. Bundle:......................... Cost:......................................
3. Bundle:......................... Cost:......................................

# Annex D: Coverage models of different transmission sites modelled by SPLAT4J

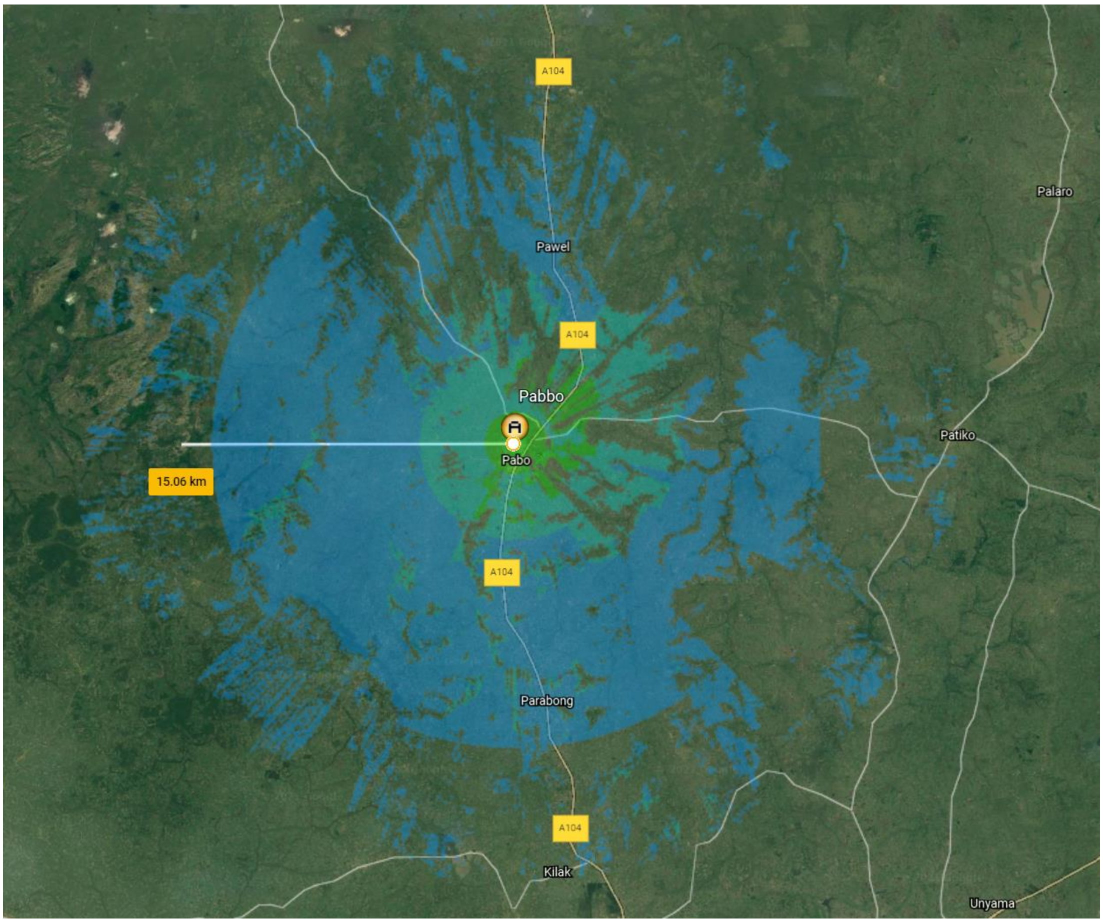



**Figure 11.1: Model for transmission at Pabbo, 15M Antenna height, 33.4W ERP, 103.8 MHz**

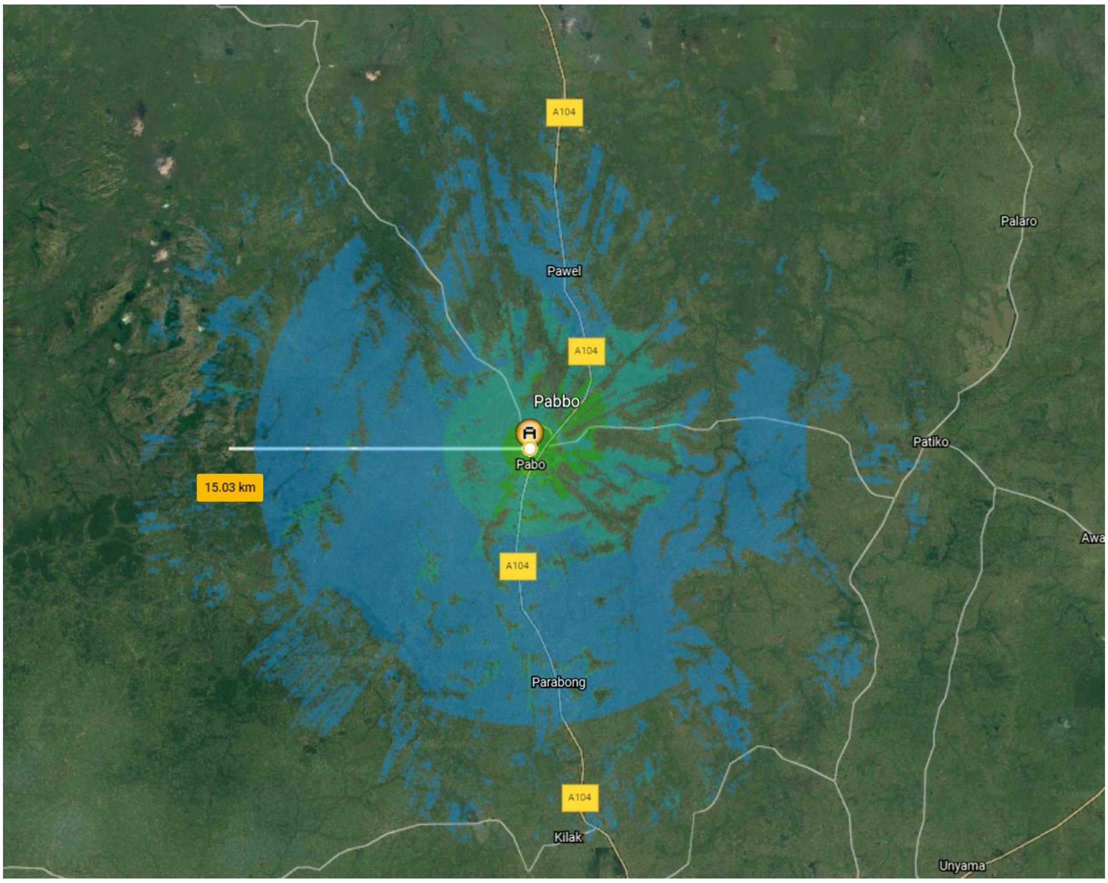



**Figure 11.2: Model for transmission at Pabbo, 15M Antenna height, 70W ERP, 103.8 MHz**

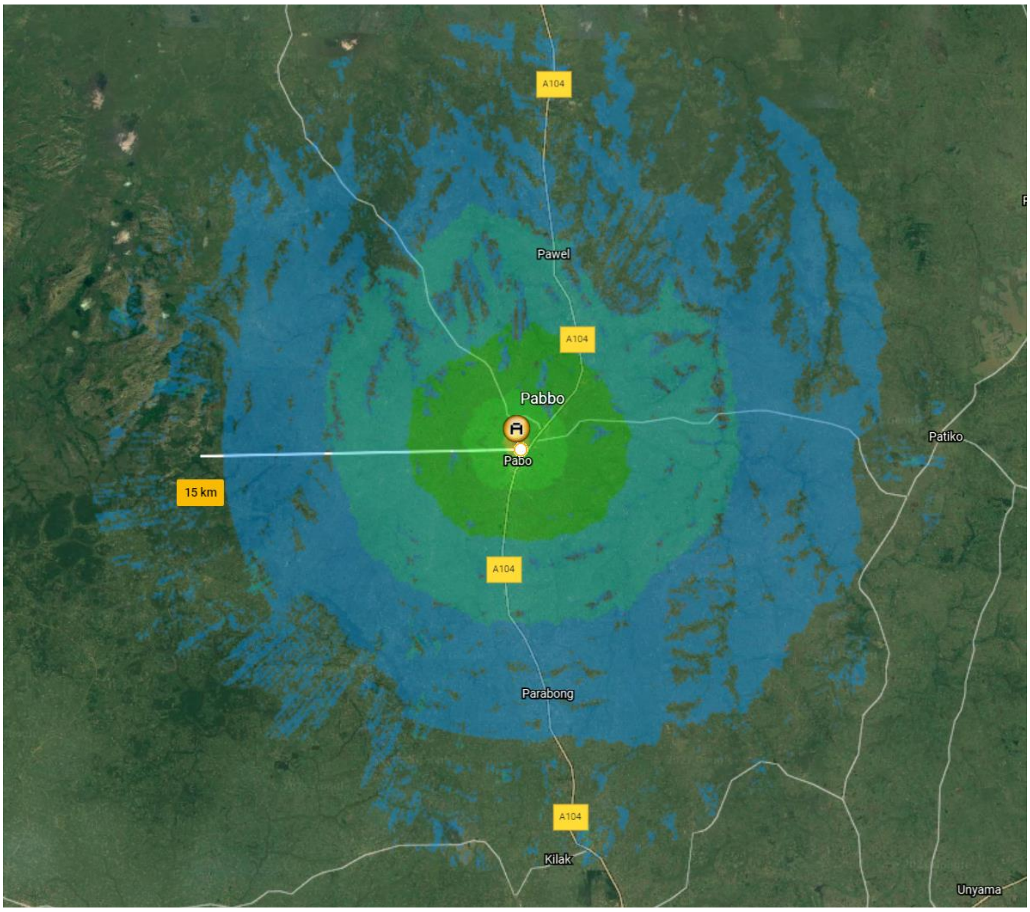



**Figure 11.3: Model for transmission at Pabbo, 50M Antenna height, 33.4W ERP, 103.8 MHz**

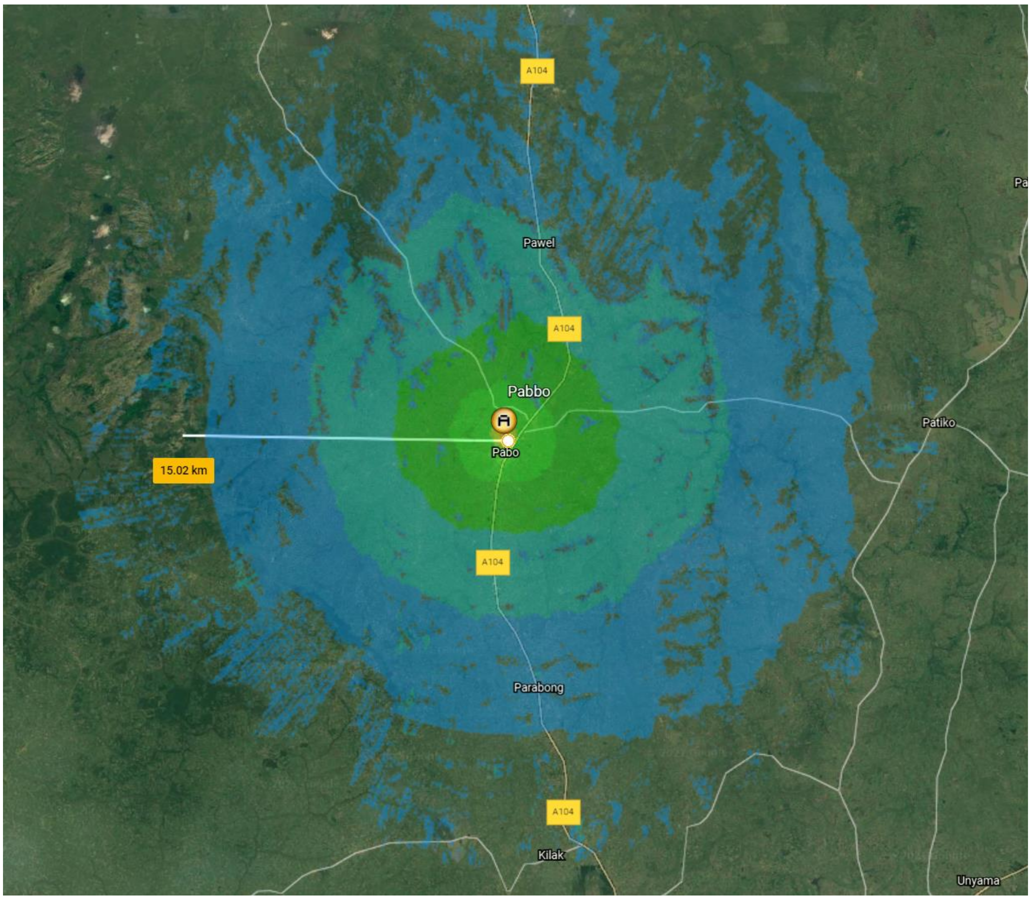



**Figure 11.4: Model for transmission at Pabbo, 50M Antenna height, 70W ERP, 103.8 MHz**

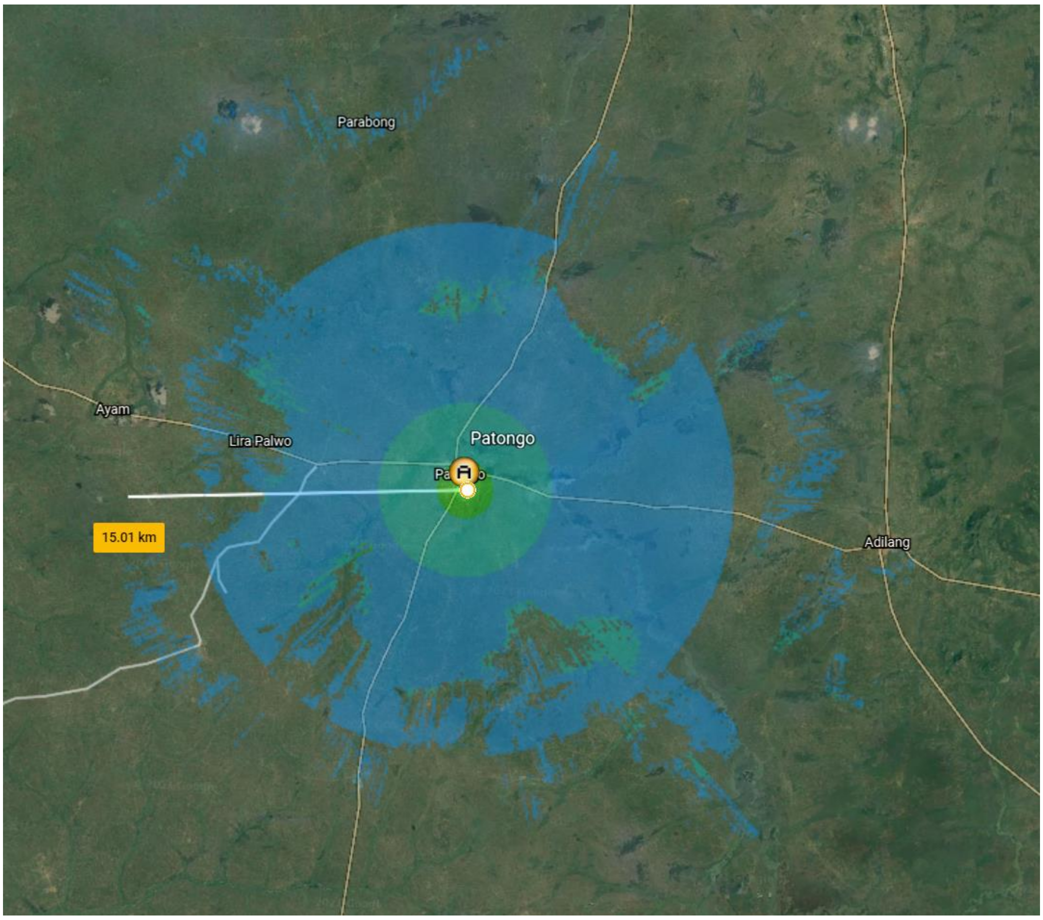



**Figure 11.5: Model for transmission at Patongo, 15M Antenna height, 24.9W ERP, 103.8 MHz**

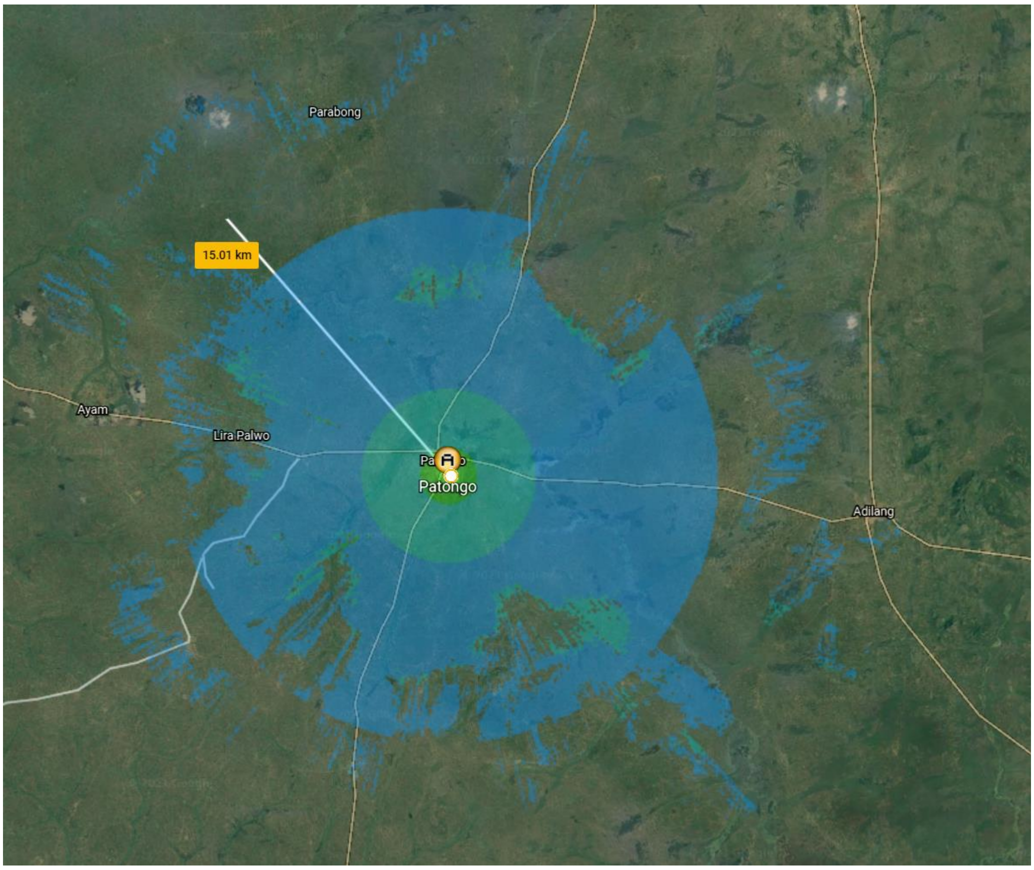



**Figure 11.6: Model for transmission at Patongo, 30M Antenna height, 50W ERP, 103.8 MHz**

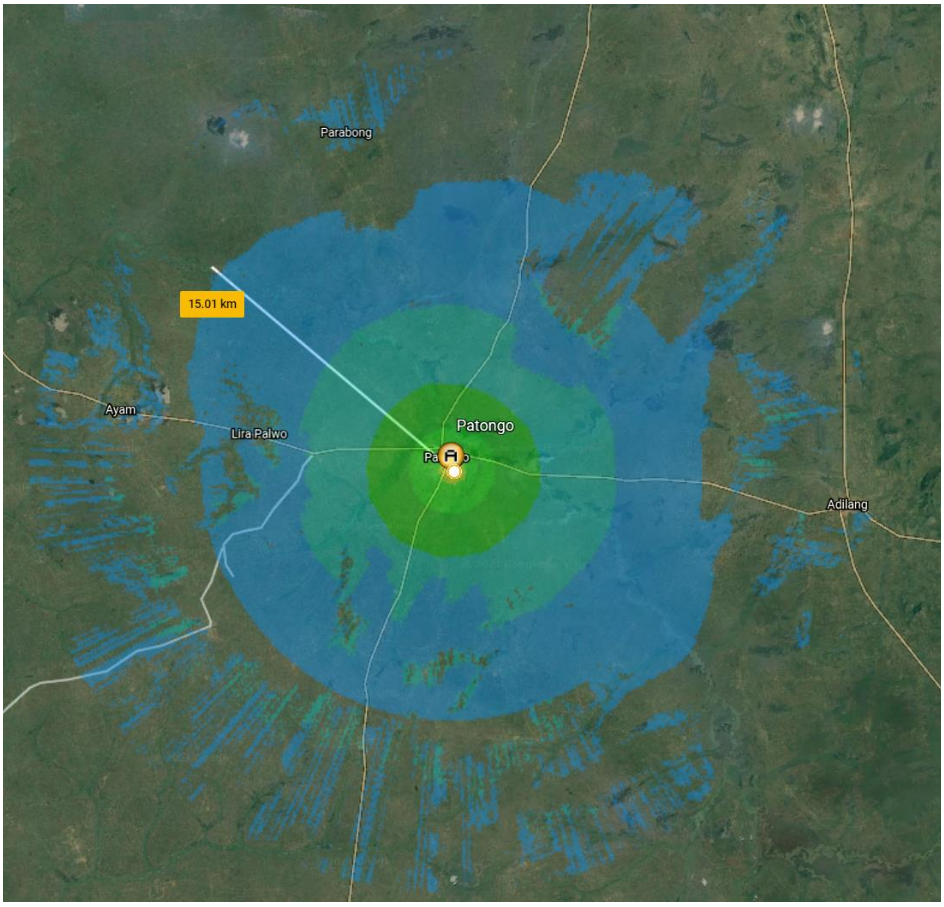



**Figure 11.7: Model for transmission at Patongo, 50M Antenna height, 24.9W ERP, 103.8 MHz**

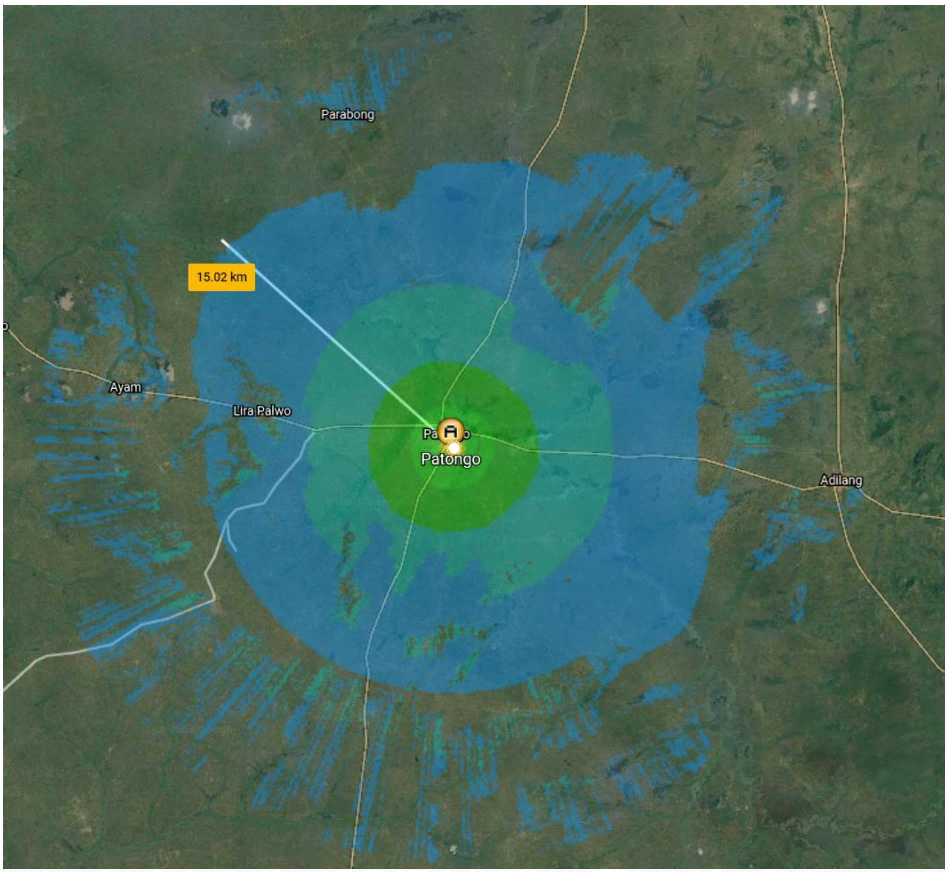



**Figure 11.8: Model for transmission at Patongo, 50M Antenna height, 70W ERP, 103.8 MHz**

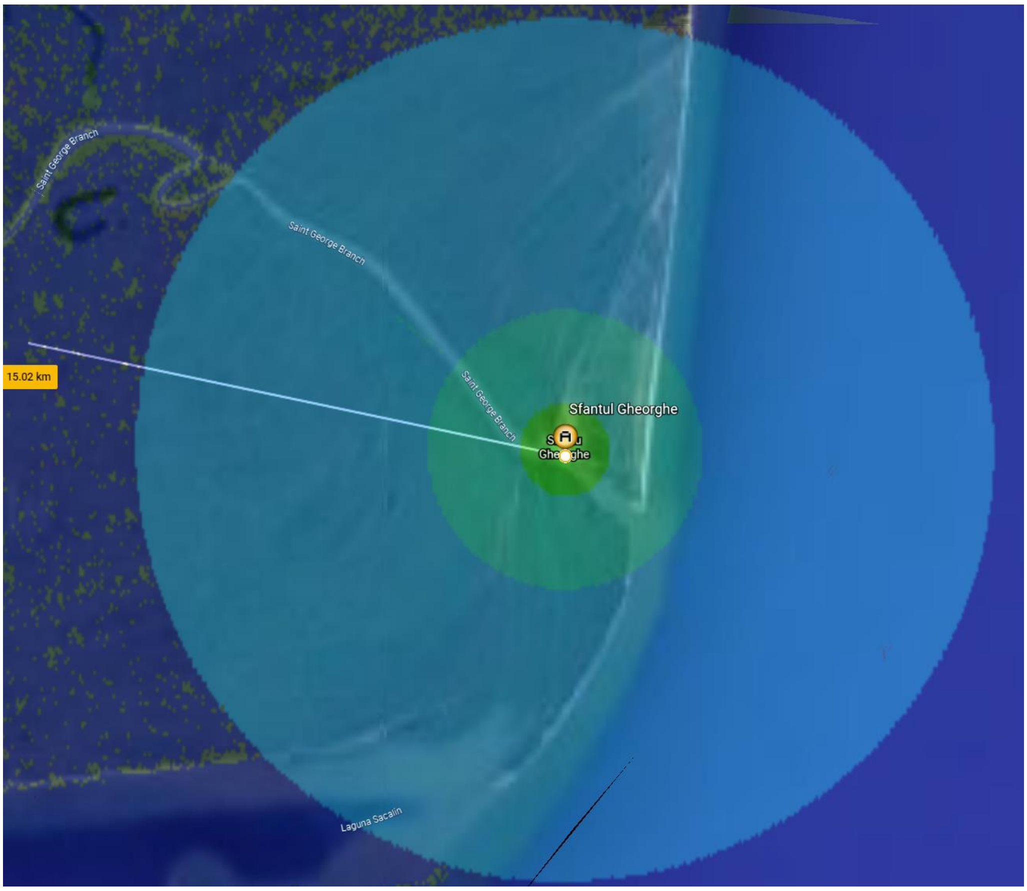



**Figure 11.9: Model for transmission at Sfantul Gheorghe, 20M Antenna height, 66.7W ERP, 106MHz**

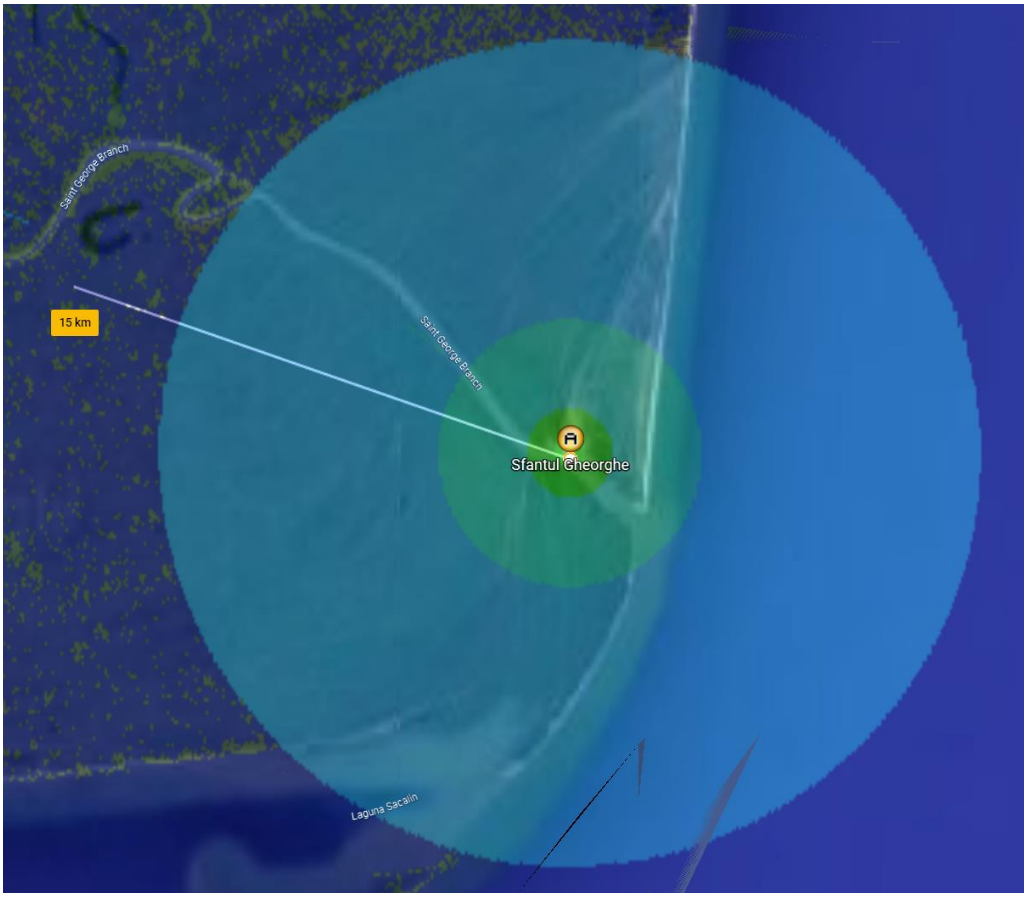



**Figure 11.10: Model for transmission at Sfantul Gheorghe, 20M Antenna height, 100W ERP, 106 MHz**

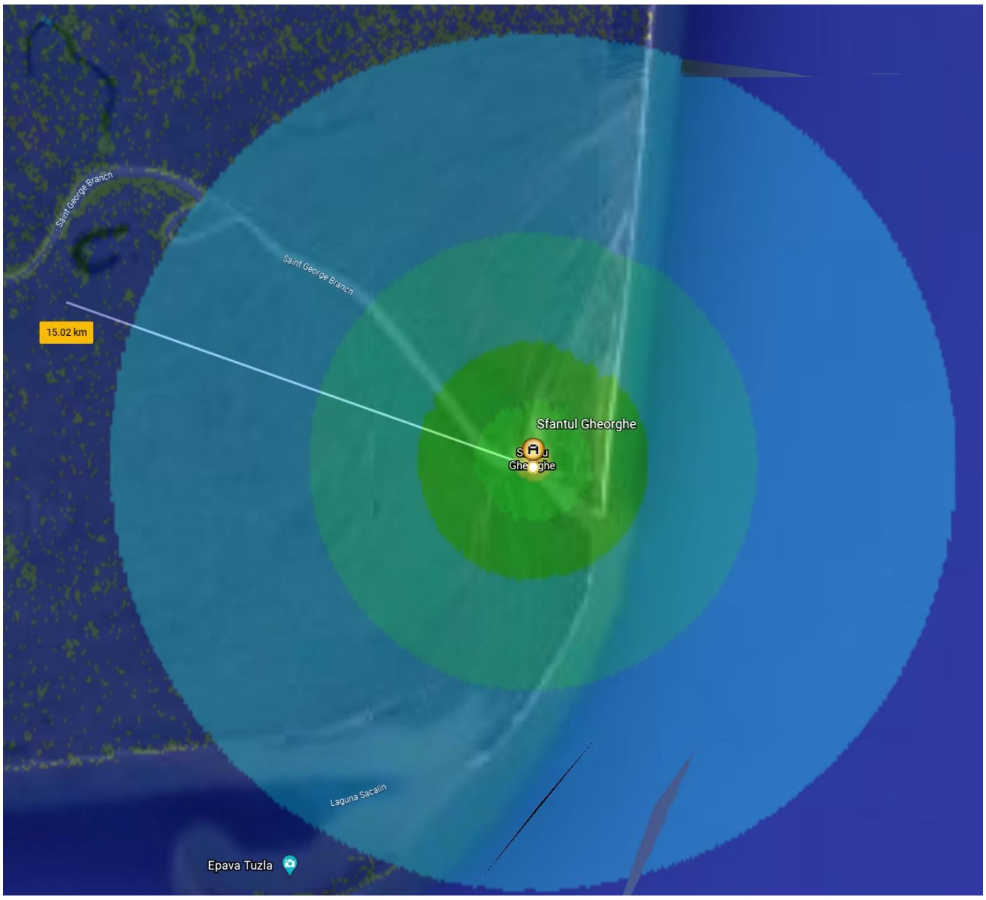



**Figure 11.11: Model for transmission at Sfantul Gheorghe, 50M Antenna height, 66.7W ERP, 106 MHz**

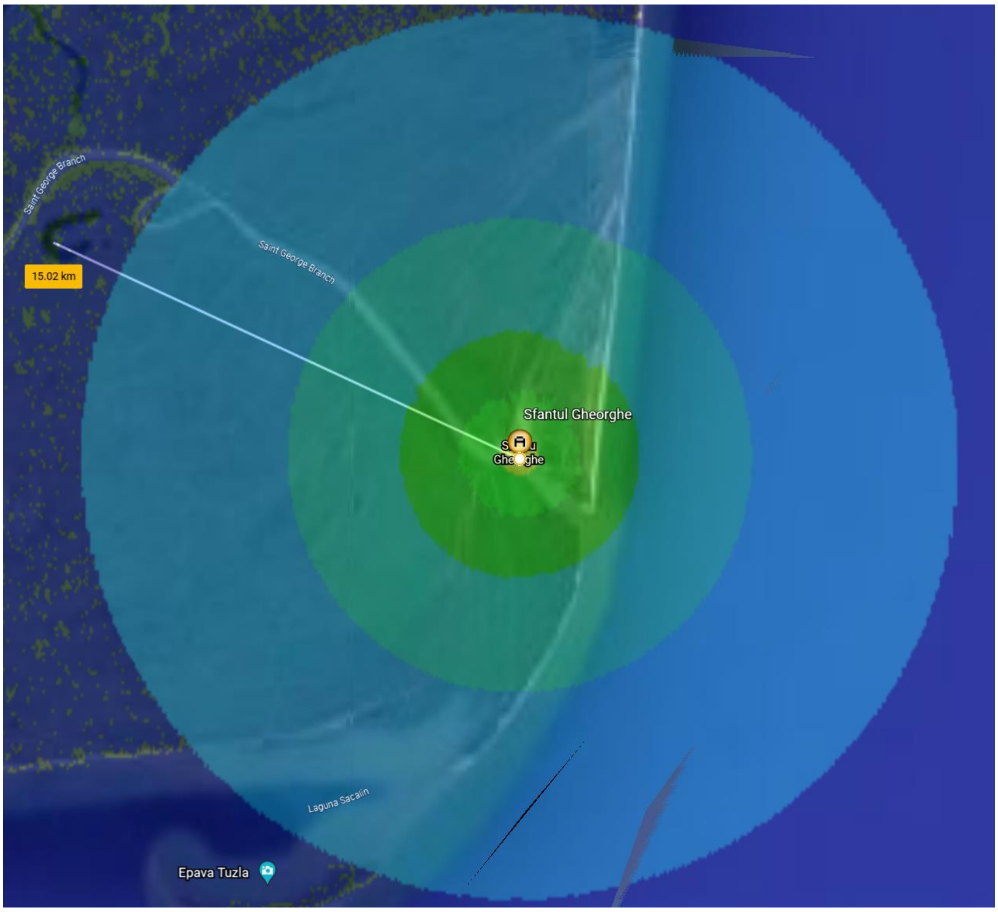



**Figure 11.12: Model for transmission at Sfantul Gheorghe, 50M Antenna height, 100W ERP, 106 MHz**