Community Telephone Networks in Africa

Bridging the gap between poverty and technology

Curtis Sahd
Department of Computer Science
Rhodes University
Grahamstown, South Africa
curtissahd@gmail.com

Hannah Thinyane
Department of Computer Science
Rhodes University
Grahamstown, South Africa
h.thinyane@ru.ac.za

***Abstract*—Many new cell phones on the market come with 802.11 enabled, along with standard Bluetooth functionality. A large percentage of working class people in South Africa typically cannot afford 802.11 enabled cell phones, and thus the most applicable form of wireless data transfer is achieved through the Bluetooth protocol. This paper investigates bridging Bluetooth and 802.11 protocols on low cost wireless routers equipped with a Broadcom chip and a Universal Serial Bus port, as well as bridging on high end cell phones. For the router component of this research, the BlueZ protocol stack will be implemented on top of the OpenWrt platform and experiments relating to the feasibility and scalability of Session Initiation Protocol voice calls between clients on the Bluetooth network and clients on the wireless mesh network will be investigated. For the cell phone component of this bridging, Java Mobile will be used as the development platform of choice, and a comparison between bridging on the cell phone and on the wireless router will be conducted, with metrics such as latency, scalability, and minimum throughput will be considered. This paper also investigates Bluetooth throughput achieved at varying distances, as well as the relationship between the average time and the average expected time with variations in the transmission unit size. This paper provides an overview of the Mobile Media Application Programming Interface, its shortcomings, and how to overcome them. This paper proposes a low cost solution to building community telephone networks in rural South Africa, through the bridging of 802.11 and Bluetooth interfaces.**

Keywords – Wireless; SIP; Community telephone networks; BlueZ; Community polling systems.

I. INTRODUCTION

The Bluetooth protocol has been around since 1994, and its primary function is to replace wires and serve as lightweight wireless implementation for data transfer. Even though most high end cell phones are equipped with 802.11, Bluetooth still serves as the primary data transfer protocol between cell phones in South Africa. Based on a survey conducted on the streets of Grahamstown, South Africa, it was discovered that most people called someone in Grahamstown or in the surrounding region on a daily basis. Currently, the only way to make phone calls, whether local or inter-town, is to make use of a fixed landline, which the vast majority of the underprivileged do not have access to, or to make use of the ever increasingly expensive mobile service providers. Paying sky high cellular network rates to make local phone calls places an enormous burden on already financially constrained rural communities. Bluetooth alone cannot be used in a full scale implementation which would enable free local phone calls. However, the combination of Bluetooth and 802.11 mesh networks could, in the context of South African rural communities, create a system which saves rural communities millions of Rand each year. A worldwide study of 24 000 participants from 35 markets (undertaken in November 2009) showed that 86% of respondents owned a mobile phone, while only 55% owned a desktop computer. Of those who owned a mobile phone, 55% used it for media purposes, and 42% used it for transferring files and made use of the Bluetooth connection [22]. Even though these statistics show the trends of people in urban areas, people in rural areas, according to the survey conducted, will most certainly make use of these technologies to reduce communication costs.

Wireless mesh networks (WMNs) are a crucial component to this research as they bridge the gap between the Bluetooth access points and provide the throughput necessary for handling all the calls/traffic. WMNs are dynamic, self-configuring networks which are designed to span large geographical areas. WMNs could therefore be used to span the geographical area of the rural community, and possibly even connect remote rural communities to one another.

This paper aims to explore inexpensive means to creating low cost community telephone networks with existing technology in rural areas. We propose a system which enables the seamless integration of Bluetooth and 802.11 on the OpenWrt and Java Mobile (JavaME) platforms. We begin with an introduction to Bluetooth and in particular, Bluetooth networking with Piconets and Scatternets. We then investigate the throughput achieved by the Bluetooth protocol at varying distances between two class two devices, followed by a comparison between the average time and the average expected time for Bluetooth transmissions with varying transmission unit sizes. We then provide a brief overview of the OpenWrt platform and focus on mesh networking, as well as reviewing related work in this area. Section IV then describes the Mesh Potato, and the possibilities it presents in rural areas. Section V introduces the Mobile Media API (MMAPI) as well as the concept of double buffering. Section VI then provides an in depth analysis of the proposed infrastructure of the Blue Bridge, and the associated advantages and disadvantages of various implementations. Section VI also provides a high level understanding of the components and techniques as well as the challenges and constraints involved in transmitting voice data from one device to another. Section VII then describes the context of this paper and how the proposed technology can be beneficial to rural communities and coincides with objectives of various social reconstruction programmes. Section VIII describes possible future works on this concept, as well as the challenges and constraints involved in implementing these extensions. Section VIII then concludes this paper.

II. Bluetooth Personal Area Networks

A. *Overview*

At initial conception Bluetooth was considered the future of Personal Area Networks (PANs), due to it being a lightweight protocol and the inexpensive manufacturing of Bluetooth chips [2]. The Bluetooth specification clearly defines PANs and associated roles of the nodes in the PAN in the case where two devices are communicating directly. The Bluetooth specification also defines the roles of nodes in multi-hop environments, but less research has been conducted in this field [2]. Asthana and Kalofonos [3] have developed a custom protocol which enables the seamless communication of existing Piconets within a Scatternet. Specifically, their research allows for the creation of Ethernet and IP local links on top of scatternets through the use of a standard PAN profile implementation, without the need for ad hoc forwarding protocols [3].

Plenty of research has been done in the field of providing Internet Access to rural communities. There has been little to no research in the field of making use of low cost hardware infrastructure to bridge Bluetooth and 802.11 which enables large scale service provision to local and remote rural communities. Bluetooth Piconets and Scatternets are an important component of bridging Bluetooth and 802.11 mesh networks, as in some cases devices will be able to communicate directly with one another (Piconets) and in other cases devices may only be able to communicate by sending traffic through a number of other nodes before reaching the desired node (more applicable to Scatternets). With that said, many researchers have investigated the formation and limitations of Mobile Ad-hoc Networks (MANETs) with the Bluetooth protocol [2].

B. *Piconets and Scatternets*

According to Bisdikian [4] a piconet is simply defined as a collection of Bluetooth devices which can communicate with one another. A Piconet consists of one master node and one or more slave nodes, and exists for as long as the master communicates with the slaves. Piconets are formed in an ad-hoc manner, and need a minimum of one master node and maximum of seven active slave nodes. Although only seven active nodes are able to transmit based on coordination of the master node, other nodes are able to connect to the Piconet, and are said to be in a parked state [4].

Scatternets are based on Piconets and are said to exist when one device is a member of multiple Piconets. In the case of Scatternets, a node can only serve as the master node for one Piconet.

For the purposes of this research it is important to understand the functioning of Piconets and Scatternets in order to handle the association of clients with Bluetooth access points.

As mentioned above, Piconets consist of a maximum of eight active nodes, consisting of the master node, and seven active slave nodes. In the context of this research as well as the broader context of community telephone networks, the participation of a maximum of seven active clients poses a very real problem in terms of scalability as well as feasibility.

Apart from the issue with scalability, another concern is the throughput achieved by the Bluetooth protocol at varying distances. Subsection C provides an overview of the relationship between throughput and distance for class two Bluetooth devices.

C. *Bluetooth throughput at distance*

A major consideration for the successful implementation of Blue Bridge is the throughput achieved at varying distances between access points and clients. Although the maximum theoretical transmission distance specified by the Bluetooth specification for class two Bluetooth devices is 10m, Sahd conducted experiments which resulted in an increase in the maximum transmission distance. Sahd [13] conducted throughput tests at 1m intervals until such point was reached where the file transfer was unsuccessful. A 6.6 MB audio file was transmitted four times with an MTU of 668 bytes, and the results averaged to achieve the results depicted in Figure 1. Sahd [13] also found that reliable transmission was feasible for distances up to 15m with class two Bluetooth devices. Figure 1 highlights the results obtained from distance variation during class two Bluetooth transmission:

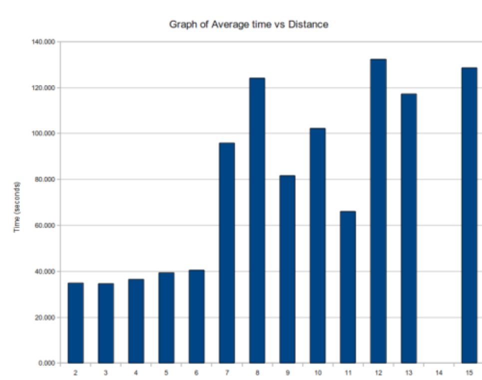

Figure 1. Graph of average time vs distance

It can clearly be seen that the maximum distance achieved by the class two Bluetooth transmitter in the Nokia N95 8GB is capable of achieving distances of up to 15m. There are no results for the transfer at the 14m mark, as link disconnection was prevalent. This could be as a result of electrical piping and/or other environmental factors which adversely affected the connection clarity at the 14m mark. An interesting observation is the throughput achieved at the 11m distance interval, which exceeded that achieved at the 7m, 8m and 9m distance intervals. The result set is theoretically inconsistent, and atmospheric and/or environmental conditions could be the only possible explanation for these differences in throughput. With an understanding of the relationship between distance and throughput for class two Bluetooth devices, Subsection D investigates throughput optimization by determining the optimum range of transmission unit sizes.

D. Relationship between transmission unit size and throughput

JavaME enables variation of the Bluetooth transmission unit, which proved to greatly influence the transfer speed, and hence the transmission distance and quality of the transmission. Although larger transmission unit sizes logically achieve the fastest transfer speed and the quality of voice/media playback, this is however not the case with audio streaming on Java Mobile devices. Sahd performed an experiment which tested the transfer speed and quality of playback by reducing the transmission unit size from 668 bytes to 67 bytes (10% decrements). This experiment shows the difference between the expected time and the average expected time, thus enabling a throughput comparison between the various transmission unit sizes. This experiment also shows the differences between theoretical Bluetooth throughput and actual Bluetooth throughput. The experiment was conducted by sending a 6.6 MB audio file from a Nokia N95 8GB to a Nokia N82. The following formula was used in calculating the average expected time:

x = time taken to transfer file using the maximum MTU
n = percentage of the maximum MTU
Expected time = (x * (100% - n%)) + x

Figure 2 shows the relationship between the average time and the average expected time for each transmission unit size while transferring a 6.6 MB file between two cell phones using Bluetooth:

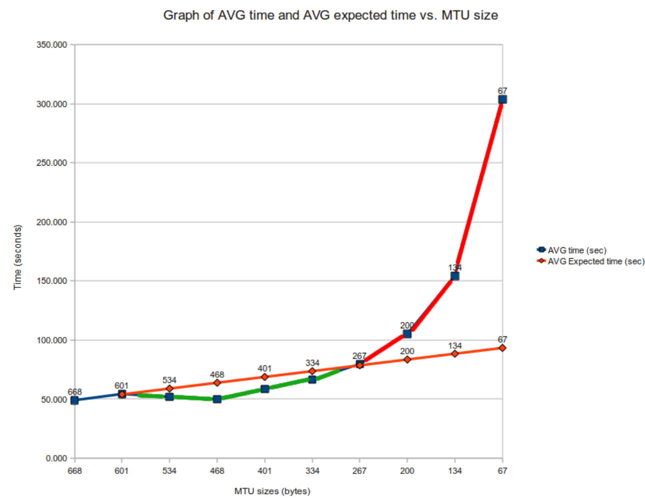

Figure 2. Graph of average time and average expected time vs. MTU size

Figure 2 shows the average time vs the average expected time for a given transmission unit size. The green portion of the graph represents the portion where the average time is less than the average expected time. The orange portion of the graph represents the case where the difference between the average time and the average expected time is greater than or equal to 0 seconds, and less than or equal to 2 seconds. The red portion of the graph represents the case where average expected time is less than the average time. From Figure 2 it can be seen that average time was less than the average expected time for transmission units 534 through 334. The average expected time however is less than the average time for transmission units 200 through 67. The two most obvious points on the graph are those of 601 and 267 where the average time is equivalent to the average expected time. With that said, the transmission units 601 and 267 are ideal for testing Bluetooth throughput which closely matches theoretical throughputs.

Section II introduced the Bluetooth protocol; the formation of PANs as well as their limitations; and the optimization of the Bluetooth protocol by means of distance and MTU variation. Section III introduces the OpenWrt platform and how it can be used in the context of mesh networking and service provision.

III. OPENWRT

OpenWrt is defined as Linux for embedded devices [5]. OpenWrt provides a plethora of opportunities for robust application development and service provision on embedded devices, and for the purposes of this research, specifically on wireless routers. In order to grasp the functioning of OpenWrt it is necessary to understand the various components of the software which manages wireless routers and for that matter any embedded device. The software which runs on computer chips or on embedded device chips is known as firmware [6]. The following are a few of the many types of chips which have firmware installed on them: read-only memory (ROM); programmable read-only memory (PROM); erasable programmable read-only memory (EPROM). PROMs and EPROMs are designed to allow firmware updates through a software update [6]. In order to compile custom Linux firmwares on embedded devices, a technique known as cross compiling is used, where a new compiler is produced, which is capable of generating code for a particular platform, and this compiler is then able to compile a linux distribution customized for a particular device [7]. Generally, the cross-compiling process begins with a binary copy of a compiler and basic libraries, rather than the daunting task of creating a compiler from scratch [7]. The remainder of this section describes mesh networking principles and practices on the OpenWrt platform, as well as the state of the art in rural mesh networks.

OpenWrt contains a number of packages which assist with the implementation of mesh networks. Optimized Link State Routing Protocol (OLSR) is an example of a routing protocol developed by Andreas Tønnesen which has been implemented in the form of a package for OpenWrt [8]. Another Open Source mesh networking implementation known as ROBIN (ROuting Batman Inside) has been developed on top of OpenWrt Kamikaze [9]. ROBIN is self-configuring and self-maintaining, which enables the seamless creation of wireless mesh networks. ROBIN requires a minimum of one Digital Subscriber Line (DSL) connection, a Dynamic Host Configuration Protocol (DHCP) enabled router which is connected to the DSL line and serves as the gateway node [5]. Client repeater nodes simply have to be powered on and a mesh network is dynamically configured [5]. With that said, open mesh networking protocols, which simplify the creation and extension of mesh networks, can be utilized in rural communities. Mesh networks thus serve as a low cost alternative to information technology service provision in rural areas, providing significantly more benefits than drawbacks. The benefits of mesh networks in rural communities have been extensively discussed [9] [10]. Reguart et al. [9] suggest that mesh networking technologies in urban areas are often unsuited to rural areas due to the high cost of equipment and maintenance. They proposed and tested Wireless Distribution System (WDS) by making use inexpensive wireless hardware (Linksys WRT54AG and Linksys WRT54G). Through a prototype deployment of their infrastructure they found that inexpensive wireless equipment is capable of providing fourty people with internet access, and at any one point in time there are between fifteen and thirty active clients [9]. The aforementioned implementation performs surprisingly well for sparsely situated rural communities, but would not suffice for the purposes of South African rural communities for the following reasons: Rural communities in South Africa are densely populated; laptops are seldom found in rural areas, as most of the people are living below the breadline and cannot afford such equipment; even if everyone had access to laptops, the use of inexpensive wireless equipment as used above would be overloaded and the end result would most likely be malfunction; also the use of secured outdoor equipment is imperative in the context of South Africa due to crime levels.

This Section provides an introduction to the OpenWrt platform, its versatility, and the benefits it provides in service provision. Section IV introduces the Mesh Potato, and how it utilizes the OpenWrt firmware.

IV. THE MESH POTATO

The Mesh Potato is a new device which merges the ideas of current telephony (analog phones) and future technology (reliable wireless communications). The Mesh Potato combines a wireless access point (AP) with an Analog Telephony Adapter (ATA), and thus enables cheap communications using existing technology [18]. Routers by Meraki [19] and OpenMesh [20] are gaining popularity due to their low cost and robustness, but they however lack the functionality contained within the Mesh Potato in terms of integration with existing telephonic infrastructure.

Although rural areas in South Africa are often on the outskirts of town, plenty of remote and isolated settlements exist, and more often than not, these settlements lack infrastructure such as running water, sewage and waste removal, and electricity. In such cases where electricity is scarce or non-existent, the Mesh Potato is ideal since it can be powered by a 10w solar panel [18].

The Mesh Potato is powered by Open Source firmware (Linux, OpenWrt, B.A.T.M.A.N and Asterisk) which removes vendor lock in and makes the Mesh Potato cost effective and highly configurable [18]. The Mesh Potato enables the seamless connection of analog telephones, as well as wired and wireless IP phones. Cellular technology is the primary form of communication in rural areas in South Africa, and although analog phones are inexpensive and could be subsidized by the government, the Mesh Potato is currently unable to cater for the existing needs of people in rural areas.

This section shows how OpenWrt can be used in service provisioning scenarios. Section V introduces the Mobile Media API (MMAPI); highlights its benefits in the realm of media streaming; and describes its shortcomings and possible solutions in overcoming them.

V. THE MOBILE MEDIA API (MMAPI)

The Mobile Media API (MMAPI) is an optional API which enables advanced multimedia capabilities on the Java enabled devices [25]. The MMAPI enables the playback of different audio and video formats from the network, from a record store (persistent storage on JavaME devices), from a JAR file, or from dynamic buffers. The MMAPI also enables audio and video capturing, as well as streaming on Java enabled devices.

One of the major problems with streaming on the JavaME platform is the lack of RTP and RTSP support. One of the more implemented methods of streaming live audio (voice transmissions) is the use of the SIP protocol for signaling; the RTP protocol for transmitted the media; and the RTSP protocol for describing the media being transmitted with the RTP protocol. Although the JavaME platform supports SIP communication, the lack of the other two aforementioned protocols renders streaming somewhat more complicated. With that said other methods of describing the media and transmitting it have to be devised.

There are three main components of the MMAPI: The Manager; the Player; and the DataSource. The Manager class is responsible for Player instantiations, which in turn sources the data from the DataSource, thus enabling playback. The Manager class essentially bridges the gap between the Player and the DataSource. Figure 3 shows the MMAPI architecture:

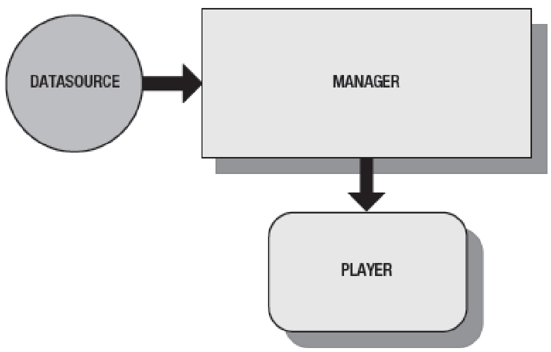

Figure 3. MMAPI Architecture

Without the MMAPI streaming on the JavaME platform would be impossible without the development of other APIs or frameworks which make it possible. The lack of widespread support for RTP and RTSP on the JavaME platform, changes the streaming from the traditional structured and proven method, to one consisting of: record; compress and serialize; and transmit. The Player class contained within the MMAPI is capable of playing media from various sources, one of which includes network streaming. The MMAPI specification however does not mention that streaming is only possible once the entire media file (audio or video) has been downloaded. According to Costello [26], streaming is defined as the process of transferring data from source to destination, where the destination device decodes the data before all the data has been transferred. With that said the MMAPI not only lacks support for streaming protocols such as RTP and RTSP, but there are also limitations with the way in which the Player class needs the entire media stream before being able to render any media playback.

Vazquez-Briseno and Vincent [27] propose a media streaming architecture consisting of a streaming server; a multicast proxy; and a mobile client. Their architecture utilizes the RTP; RTSP; and Session Description Protocol (SDP) protocols to stream an AMR audio file from the streaming server to the mobile client. Their client made use of the MMAPI, and more specifically, the Player class which played back the audio before the entire media file had been downloaded/streamed. The technique which they used to accomplish this partial media playback, is known as double buffering.

As such we applied the same double buffering technique to overcome the limitations of the Player class in the MMAPI. Table I shows the double buffering technique involved in streaming media on the JavaME platform:

TABLE I. AUDIO STREAMING WITH DOUBLE BUFFERING TECHNIQUE

Time	Buffer 1	Buffer 2	Player 1	Player 2
T_1	Receiving			
T_2		Receiving	Play Buffer 1	
T_3	Receiving			Play Buffer 2
T_4		Receiving	Play Buffer 1	

From Table I it can be seen that the double buffering technique starts with buffer 1 receiving part of the media stream at T_1. Once there are enough bytes to begin playback (normally around 150 KB), buffer 2 then receives the next part of the media stream, and player 1 then plays the bytes contained in buffer 1. At T_3 the bytes in buffer 1 are then replaced by the next part of the media stream, and the bytes contained in buffer 2 are then played using player 2. This process of alternating buffers and players then continues until the end of the media stream is reached.

Even though the double buffering technique is the only possible way of streaming on the JavaME platform short of developing RTSP and RTP streaming functionality at the application level, this technique has its fair share of disadvantages. One of the most obvious disadvantages is the jitter encountered during playback between the alternating players. The time it takes for the mobile phone to switch from one player to the other is negligible, but enough to cause a slight delay in handing over playback to the other player.

This section introduced the MMAPI and its role in media streaming. This section also showed how the MMAPI can be adapted in order to cater for the lack of support for certain protocols on the JavaME platform. Section VI describes the proposed infrastructure for bridging the 802.11 and Bluetooth protocols by means of the JavaME and OpenWrt platforms. Section VI also investigates streaming on the JavaME platform, and highlights the challenges and constraints, as well as possible solutions. Section VI also provides a costing analysis of the proposed infrastructure, as well as a means of funding.

VI. PROPOSED BRIDGING INFRASTRUCTURE

After extensive literature reviews we found that there is a lack of knowledge in the field of Bluetooth and 802.11

bridging in the context of rural communities in Africa, and as such we propose a system (Blue Bridge) which not only deals with remote access to such communities, but also enables service provision through the use of inexpensive and readily available technology thus connecting the unconnected. The system will be centered around the OpenWrt firmware, which is to be installed on the Ubiquiti AirRouter [11]. The AirRouter will not only serve as an interface for 802.11 connections, but will also become a Bluetooth access point through the use of the BlueZ protocol stack which controls the functioning of the Bluetooth dongle inserted into the USB port of the AirRouter. Asterisk [12] will be installed as a package on the OpenWrt platform, and will serve as the SIP controller. A package will be developed for the OpenWrt platform which will bridge the connections between the 802.11 and Bluetooth interfaces. Figure 4 shows the proposed infrastructure involving one AirRouter:

Figure 4. Proposed OpenWrt infrastructure for low cost community telephone network.

Any cell phone on the Bluetooth interface of the Blue Bridge should be able to place SIP calls to any other phone on the Bluetooth interface, as well as to any phone on the 802.11 interface. Of course as mentioned in Section B a maximum of seven active connections can exist on the Bluetooth interface, which clearly places limitations on the scalability of the proposed system.

With the aforementioned, the components of the proposed system include the AirRouter (running OpenWrt); the USB Bluetooth dongle; and a JavaME enabled cell phone, which the majority of the surveyed population possesses. The aim of this research is to provide Bluetooth access (via the connected Bluetooth USB dongle) as well as 802.11 access to multiple geographically dispersed routers which in turn enables the creation of community telephone networks, thus connecting the unconnected, and significantly decreasing the burden of expensive cellular calls.

The ideal scenario is the use of minimal equipment, while still maintaining an acceptable level of service provision. This translates to decent quality voice calls, with minimal downtime. In order to achieve this, an understanding of the Bluetooth protocol and its scalability limitations is vitally important. Sahd [13] conducted a study which investigated the real throughput achieved by the Bluetooth protocol on mobile devices. Sahd [13] found that the average transfer speed of the Logical Link Control and Adaptation Protocol (L2CAP) when transferring a 6.6 MB M4A audio file twice between two cell phones is 136.39 KBps [13]. If a maximum of seven clients are connected to the Bluetooth interface each client would be allocated a bandwidth of 19.48 KBps. Based on the assumption that seven simultaneous connections are active on the Bluetooth interface, the minimum accumulated bandwidth for these connections is 27.35 KBps, which would allow a theoretical number of thirty five clients to be connected [14].

This research will also investigate the differences in performance of Blue Bridge implementations on the JavaME platform and on the OpenWrt platform. Of course the most prominent difference between implementations on the two platforms is the class of Bluetooth device. The OpenWrt platform implementation of the Blue Bridge will make use of a class one Bluetooth device which is capable of a distance of 100m, whereas cell phones typically contain class two Bluetooth chips which enables transmission at distances of 10m. There are three classes of Bluetooth of which class three achieves a distance of up to 1m [23]. Sahd [13] found that even though the Bluetooth specification states a distance of 10m, transmission is possible at distances as high as 15m. Figure 5 shows the proposed Blue Bridge infrastructure on the cell phone:

Figure 5. Mobile phone based implementation for community telephone networks.

From Figure 5 it can be seen that an external asterisk server would have to substitute the asterisk server contained within the OpenWrt packages. The scalability of the internal asterisk server would have to be researched and compared to that of the external asterisk server. On the other hand, the Nokia N95 8GB could pose to be a severe bottleneck under load.

In order to determine which platform will serve as the basis for a community telephone network, a number of metrics would have to be compared. These metrics can be seen in Table II:

TABLE II. KNOWN METRICS OF PROPOSED BLUE BRIDGE PLATFORMS

Metrics	OpenWrt	Blue Bridge on cell phone
Cost	Cheap	More expensive
Compactness	Average	Very compact
Complexity	High	Medium
Platform	Linux	Java Mobile

Based on the information currently available, assumptions from the data in Table II could lead one to believe that the Blue Bridge on the cell phone would be the better alternative as a whole. However, metrics such as performance under load, scalability, and multi-hop capability can only be determined once the implementation and necessary research has been completed.

Even though Figures 4 and 5 depict the bridging infrastructure necessary for connecting previously disconnected Bluetooth devices, one crucial component is that of interconnecting the bridging nodes at varying distances. Ideally, the interconnection of these nodes should be extendable from one rural community to another, possibly for distances above 50km.

With the above overview of the equipment needed for the implementation of the Blue Bridge, Subsection A provides information pertaining to the techniques used to transmit and receive audio transmissions. Subsection B highlights the costs involved, and a means for funding the Blue Bridge.

A. *JavaME streaming infrastructure*

As mentioned in Section V., streaming on the JavaME platform involves the adaptation of existing classes and APIs. This section deals with the techniques involved in successful implementation of the client side application for this research.

Due to JavaME supporting the SIP protocol, and its trivial implementation on this platform, SIP has been omitted from the diagrams and can be assumed present. Figure 6 shows the interaction between two mobile phones participating in an audio call, and the procedures involved in ensuring its success:

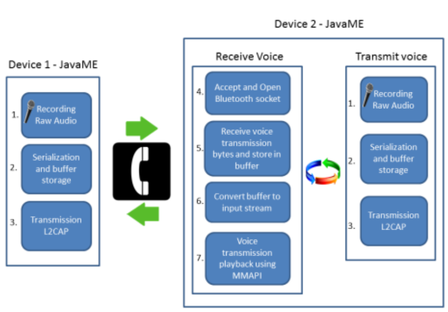

Figure 6. Sequence of events for mobile phone based implementation of community telephone networks.

As mentioned in Section V., the MMAPI lacks support for the major streaming protocols on the JavaME platform. From Figure 6, it can be seen that Device 1 initiates the call, and the procedures necessary in doing so are: Recording the audio via the MMAPI into a temporary buffer, from which the audio data is streamed to Device 2. In order for Device 2 to accept the voice call and receive the subsequent audio transmission, an *accept and open* method is continually running, ready for any incoming transmissions. With the inability of the Player class in the MMAPI to play partially downloaded media, it is necessary to store the incoming media in a temporary buffer, and play back the audio from a static buffer. A static buffer in this context is a buffer which contains media content which does not change, thus fooling the Player class into thinking that the media contained in the buffer is in actual fact an entire media file which is not being streamed. As described in Section V., once the media contained in the static buffer is rendered/played back, the media from the temporary buffer is then transferred to the static buffer, thus enabling constant media playback.

Of course throughout this process of receiving the streamed audio, storing it in buffers, and playing it back, Device 2 might also want to transmit audio back to Device 1. With that said, this then gives rise to an even greater problem, which is intrinsic to the Bluetooth protocol itself, and that is the fact that transmission using the L2CAP protocol is half-duplex. This translates to only one device/party being able to transmit at a time, essentially creating a push-to-talk system which can by no means be considered a phone call.

Clearly, when compared to full-duplex communication, which is that of traditional phone calls, this system is limited. However in a country such as South Africa where poverty is so widespread, the benefits of being able to communicate for free, far outweigh the limitations of half-duplex communication. In the context of instant messaging and community polling, half-duplex communication is completely acceptable.

B. *Costs and implementation considerations*

There are two important factors to consider when determining the cost, and the number of units necessary for the implementation of community telephone networks: the geographical area and the proposed number of connected clients. The geographical area plays a large role in determining the strength of the devices needed to transmit a good quality signal. Mountains, trees, buildings, and other obstructions have to be considered. The number of connected clients dictates the scalability of the system, and thus the overall cost of implementation. Table III provides an overview of the costs involved:

TABLE III. KNOWN METRICS OF PROPOSED BLUE BRIDGE PLATFORMS

Device	Cost	Means of funding
AirRouter 150Mbps WiFi Router	R313.50	Government
Mecer Class 1 USB Bluetooth (ENUBT-C1EM)	R169.00	Government

Device	Cost	Means of funding
Basic machine for Asterisk server (1.8 GHz, 2GB RAM, 500GB HD)	R2700.00	Government

Based on the costs in Table III, the maximum total cost for a prototype system catering for seven connected nodes will come to a total of R3182.50. This value is of course inclusive of the Asterisk server machine, which would not be necessary if the Asterisk server were to be implemented on the AirRouter itself.

The average voice call from the Vodacom cellular network to another network costs R2.75 per minute [21]. Based on the assumption that seven people spend five minutes on the phone each day for one month, the total cost incurred is R2983.75. Even though the Bluetooth protocol only permits seven active clients, more than seven people could connect to one AirRouter, due to the unlikeliness of everyone placing calls simultaneously. With that said, it can be seen that in just one month, the costs incurred by impoverished communities can be drastically reduced. This rate is the highest rate per minute rate on the Vodacom prepaid plan, and was chosen to estimate the maximum amount of money spent on cell phone calls.

Section VII provides an overview of government initiatives to introduce equality in impoverished areas, as such all equipment and implementation costs would be government subsidized.

VII. CONTEXT

The reconstruction and development program (RDP) of South Africa is a program implemented by the African National Congress (ANC) which addresses socioeconomic problems which exist as a result of the Apartheid regime [15]. The RDP program is of great benefit to all South Africans and in particular, South Africans living in rural areas without basic necessities such as adequate housing, water and electricity. Traditionally RDP housing was built on plots of $250m^2$ which placed tremendous strain on the fair land distribution due to special constraints [16]. Recently, there has been a movement from traditional RDP housing to more cost effective multi-storey RDP housing which reduces plot sizes from $250m^2$ to $80m^2$ [16][17]. With that said this poses as an ideal situation for the successful implementation of the Blue Bridge, as signal penetration will be higher and this type of RDP housing would prove more effective from a point of view of device mounting as well as line of sight access for surrounding residents. The Blue Bridge will benefit such communities immensely in terms of cost savings, and possible expansions could include educational resources and Internet access.

Section VIII concludes this paper and provides possible extensions to this research.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we proposed an inexpensive means to creating a community telephone network, which utilizes existing technology and infrastructure. We demonstrated an innovative approach to merging two independent technologies to achieve maximum penetration in all spheres of society. We proposed an infrastructure for the implementation of the Blue Bridge on the OpenWrt platform, as well as on the JME platform, and determined the metrics necessary for large scale implementation. This paper demonstrated an understanding of the social inequality and the effects of overpriced communications on impoverished communities.

The Mesh Potato lacks functionality which caters for the existing needs of people in rural areas. Similarly, the Blue Bridge lacks the functionality of providing an analog telephony interface, which is still widely used. As such, future work which adds functionality to the OpenWrt component of the Blue Bridge could involve connecting the Mesh Potato to the AirRouter via cable, and ensuring that both devices are on the same subnet, thus enabling the utilization of the analog interface of the Mesh Potato. In terms of the cell phone component of the Blue Bridge, the Mesh Potato could be connected to the cell phone via the wireless interface.

Another proposal for future work regarding this research could involve the use of low cost, high powered wireless equipment which could solve the need for large numbers of AirRouters or similar devices, since one device could provide access to a larger area. Future implementations of the aforementioned could involve connecting powerful wireless equipment to the AirRouter via the LAN interface, and in the case of the cell phone based Blue Bridge, via the wireless interface. The proposed expansion of the original infrastructure can be seen in Figure 7:

Figure 7. Proposed wireless expansion of OpenWrt based Blue Bridge.

Figure 7 shows the expansion of the OpenWrt based infrastructure through the use of an external high powered wireless device, which is connected to the AirRouter via cable. This device then expands the wireless network, which then enables a larger number of clients to connect to the mesh and reap the benefits of a community telephone network. Of course the AirRouter will still serve as an access point for nearby 802.11 and Bluetooth clients.

Figure 8. Proposed wireless expansion of cell phone based Blue Bridge.

Figure 8 depicts the expansion of the cell phone based infrastructure for the Blue Bridge. Since the cell phone is unable to connect to the external wireless device via LAN cable, a connection needs to be made wirelessly. As such, the external high powered wireless device will transmit signal over a greater distance accomplished by the cell phone and will serve as the primary AP for 802.11 based clients.

Blue Bridge has a plethora of benefits in the context of South Africa and its diverse demographic spread. Not only does this research provide a platform for cheap local calls, but it also gives birth to instant messaging applications, community polling systems, and possibly even crime reporting and emergency response.

An instant messaging expansion of Blue Bridge not only provides an alternative to the proposed voice communication, but also provides a means to conserve precious bandwidth. With the introduction of instant messaging capability to Blue Bridge, other possibilities arise, such as text-based community polling systems and social networks.

Although community polling systems have been researched in the context of e-voting and similar applications, the use of such systems (and their implementation) as a means for service delivery are largely unexplored. With somewhat limited resources and often overburdened government departments, the reporting and subsequent implementation and follow-up of services proves to be a very real problem in South Africa. Future uses for this research could implement community polling systems as a means to report, attend to, and track progress of issues relating to refuse removal, road maintenance, security risks and other vital services which the community depends upon. Members of the community will use existing cell phones to access the community network through a series of Bluetooth and 802.11 access points, thus enabling people with less modern cell phones to connect to the community network, post new service tickets and view current service tickets. Any community member wanting to participate in community polling system will register with their ID number through the web-browser on their mobile device, thus eliminating potential "prank service requests". Another means of reducing the frequency of illegitimate requests and hence the overall burden on the system is to only allow people above the age of 21 to register.

Community polling systems are a great example of how Blue Bridge can be used to alleviate pressure on government departments, and essentially provide the community with a means of ensuring that municipalities are aware of issues in the surrounding area. Such systems also provide a more managed approach for government departments to track open service tickets as well as a means of informing the community of pending changes and/or service disruptions.

According to Kumar and Sinha [24], e-Government is defined as the set of technological tools which are used in enhancing the functioning of government to better serve its citizens. With the wide scale adoption of Blue Bridge on mobile devices, government officials and citizens could improve the efficiency of municipalities. One of the major concerns in the implementation of such systems in a country such as South Africa where the literacy rate pales in comparison to first world countries, is of course the inability to use text-based systems. In such cases, service requests would have to be tracked by means of pictorial interfaces.

Acknowledgment

I would like to thank my sponsors, the Telkom Centre of Excellence at Rhodes University, funded by Telkom SA, Business Connexion, Verso Technologies, THRIP, Stortech, Tellabs and the National Research Foundation.

References

[1] Sahd, C., Thinyane, H., "Connecting the Unconnected," The Fourth International Conference on Advances in Mesh Networks, Mesh 2011.

[2] Zaruba, G.V., Basagni, S., and Chlamtac, I., "Bluetrees-Scatternet formation to enable Bluetooth-based ad hoc networks," IEEE International Conference on Communications, ICC 2001, vol. 1, 2001, pp. 273-277.

[3] Asthana, S. and Kalofonos, D., "Secure ad-hoc group collaboration over bluetooth scatternets," Applications and Services in Wireless Networks, 2004. ASWN 2004. 2004 4th Workshop, pp. 199-124.

[4] Bisdikian, C., "An overview of the Bluetooth wireless technology," IEEE Communications Magazine, vol. 39, 2001, pp. 86–94.

[5] OpenWrt. Available at: http://openwrt.org, 2011. [Accessed 04-04-2011].

[6] Apple. What is Firmware?. Available at: http://support.apple.com/kb/ht1471, 2008. [Accessed 01-04-2011].

[7] Fainelli, F., "The OpenWrt embedded development framework," 2008.

[8] OpenWrt. Available at: http://wiki.openwrt.org/inbox/mesh.olsr, 2011. [Accessed 06-04-2011].

[9] Reguart, A., Cano, J.C., Calafate, C.T., and Manzoni, P., "Providing Internet Access in Rural Areas: A Practical Case Based on Wireless Networks," The 2006 IFIP WG 6.9 Workshop on Wireless Communications and Information Technology in Developing Countries (WCIT 2006), 20-25 August 2006, Santiago, Chile.

[10] Parikh, T.S., and Lazowska, E.D., "an architecture for delivering mobile information services to the rural developing

world," Proceedings of the 15th international conference on World Wide Web, 2006, pp. 791-800.

[11] UBNT. Available at: http://ubnt.com, 2011. [Accessed 04-04-2011].

[12] Asterisk. Available at: http://asterisk.org, 2011. [Accessed 07-04-2011].

[13] Sahd, C. (2010). "Bluetooth Audio and Video Streaming on the J2ME Platform." Unpublished master's thesis, Rhodes University, Grahamstown, South Africa.

[14] AsteriskGuru. Available at: http://www.asteriskguru.com/tools/bandwidth_calculator.php, 2011. [Accessed 09-04-2011].

[15] Metagora. Reconstruction and Development Programme (RDP) of South Africa. Available at: http://www.metagora.org/training/encyclopedia/rdp.html, 2006. [Accessed 10-04-2011].

[16] Alexandra. Another RDP first from the Alexandra Renewal Project. Available at: http://www.alexandra.co.za/05_housing/article_0610_rdp_housing.htm, 2006. [Accessed 09-04-2011].

[17] Joshco. Sol Plaatje. Available at: http://www.joshco.co.za/solplaatje.html, 2011. [Accessed 10-04-2011].

[18] VillageTelco. Mesh Potato. Available at: http://www.villagetelco.org/mesh-potato/, 2011. [Accessed 11-04-2011].

[19] Meraki. Meraki. Available at: http://meraki.com/, 2011. [Accessed 11-04-2011].

[20] Open-Mesh. Open-Mesh. Available at: http://www.open-mesh.com/, 2011. [Accessed 11-04-2011].

[21] Vodacom. 4U Prepaid. Available at: http://www.vodacom.co.za/vodacom/Deals/Prepaid/Prepaid+Price+Plans/4U+Prepaid, 2011. [Accessed 23-05-2011].

[22] Curtis, S. Global telecoms insights 2010 focus report. Available at: http://www.tnsglobal.com, 2010. [Accessed 08-04-2011].

[23] NOKIA. Bluetooth technology overview. Available at: www.forum.nokia.com, April 2003. [Accessed 09-04-2011].

[24] Kumar, M. and Sinha, O., "M-Government – Mobile Technology for e-Government," Proceedings of the 5th international conference on e-governance. ICEG 2007. Hyderabad, India, 2007.

[25] Goyal, V., "Pro Java ME MMAPI: Mobile Media API for Java Micro Edition," Apress, 2006.

[26] Costello, S. "What is streaming?." Available at: http://ipod.about.com/od/glossary/g/streamingdef.htm, 2010. [Accessed 09-04-2011].

[27] Vazquez-Briseno, M., and Vincent, P. "An Adaptable Architecture for Mobile Streaming Applications," IJCSNS 7, 9 (2007), 79.