Bluetooth Proxying and Communication with 802.11 Wireless

An Android solution to an African problem

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*—Cellular technology has drastically simplified the way in which we communicate with one another. People are far more mobile than before, and yet just as accessible. Cellular technology plays a vital role in connecting remote areas in Africa to one another, and even in and amongst extreme levels of poverty, this technology is still employed. The major problem with cellular networks is that they are expensive to setup, and expensive to maintain, which results in consumers (often those who are living below the breadline) having to pay exorbitant rates for voice communication which often takes place in the bounds of a local community or city. This paper proposes the design and implementation of a free community telephone network in rural areas in South Africa, using low cost, yet efficient technology to fulfill the role of cellular networks. This paper shows how a single high powered Android device (*Blue Bridge*) can combine the Bluetooth and 802.11 wireless protocols using a lightweight and non-broadcasting based protocol (*Linkage*) to inform nodes on either network of one another. This paper also proposes the expansion of 802.11 wireless networks, and thus remote connectivity between *Blue Bridge* nodes through SSID and channel matching.**

Keywords-Bluetooth bridging; Bluetooth proxying; 802.11 wireless expansion; Mesh; Community telephone networks

I. INTRODUCTION

Bluetooth is a lightweight, short distance wireless protocol which was introduced by Ericsson in 1994 [1]. The Bluetooth protocol has slowly but surely been replaced by 802.11 wireless [23] on cell phones, due to its bandwidth and scalability limitations. The Bluetooth Special Interest Group (SIG) has introduced a number of improved Bluetooth versions since its inception in 1994. According to Wi-fi Planet [2] it is however unlikely that upgraded Bluetooth versions will replace 802.11 due to the current widespread use of 802.11 in existing handsets, and the significant difference in the available bandwidth between the two protocols.

With that said Bluetooth V3.0 High Speed has vastly improved data rates and according to Gsmarena [3], this new version of Bluetooth is more suited to establishing quick hassle free connections between cell phones and computers for file transfer. Gsmarena suggests that 802.11 wireless is a more suitable option for permanent and reliable network connectivity [3].

There is an ongoing debate as to which protocol should be the primary protocol for data transfer on cell phones, and although the common outcome is that 802.11 wireless is far superior to Bluetooth, this paper aims to demonstrate the importance of the Bluetooth protocol and the role it plays in impoverished communities in South Africa and the African continent as a whole.

It is no secret that poverty is widespread throughout the African continent, with the average income available for the purchase of normal to high end 802.11 wireless enabled phones dwindling in comparison to that of first world nations. As such, the vast majority of cell phones in use in South Africa do not contain 802.11 wireless technology, leaving the Bluetooth protocol as the primary option for data transfer between cell phones.

Although Bluetooth is the most widely spread protocol on cell phones in South Africa, 802.11 wireless is by no means worthless. Rural communities in South Africa are typically separated by large distances which makes the interconnection of these communities an ideal platform for the implementation of low cost 802.11 wireless technology. As mentioned above, Bluetooth is more suited toward creating an effortless wireless link between computers and cell phones for the purposes of data transfer, and 802.11 wireless is more suited to the establishment of reliable more scalable networks. As such 802.11 wireless technology is a fantastic alternative, both in terms of the cost of implementation and the scalability when compared to the already widely adopted radio based cellular networks.

The obvious challenge in this instance is the development and successful implementation of a scalable platform which combines the Bluetooth protocol and its more powerful counterpart, 802.11 wireless.

The infrastructure and methods of implementation presented in this paper are from a theoretical perspective, and as such there are no results. We provide a number of different methods of implementation and highlight existing and possible issues as well as means to overcome these constraints.

As such, this paper aims to provide an overview of an application level platform which combines Bluetooth and 802.11 wireless, through Bluetooth Proxying as well as the subsequent expansion of 802.11 wireless networks through

 ISBN: 978-1-61208-210-3

SSID and channel matching. We chose to use Android as the proof of concept platform, even though Java Mobile is the most widely spread mobile platform suited to mobile application development in South Africa. The reason for Android being the preferred development platform is the associated powerful hardware which is required for the purposes of this research. The Android platform is also more suited to wireless hotspot configuration and the implementation of audio compression methods.

Section II provides an overview of the Bluetooth and 802.11 wireless protocols. Section III gives a brief introduction to the Android platform and provides an overview of the Bluetooth and 802.11 wireless protocols on this platform. After the aforementioned background on the protocols and platforms, Section IV introduces the concept of Bluetooth proxying, the proposed infrastructure to bridge Bluetooth and 802.11 wireless on the Android platform. Section V provides a brief overview of 802.11 wireless expansion which is then followed by an overview of the benefits, constraints, and overall feasibility of a proposed large scale implementation, as well as possible future extensions to this paper in Section VI. Section VII concludes this paper.

II. Bluetooth and 802.11 Overview

A. *Bluetooth Overview*

Bluetooth was initially invented to be a cable replacement technology for mobile devices and desktop computers alike. The Bluetooth protocol was designed with three key features in mind: small chip size; very little power consumption; and low cost. With the successful implementation of the above pre-requisites for the Bluetooth protocol, it was envisioned to become the standard for seamless short range wireless communication between devices [4]. There are three power classes of Bluetooth devices which accomplish various distances and are intended for varying purposes: Class 1, Class 2, and Class 3. Class 1 Bluetooth devices achieve a range of 100m, Class 2 devices achieve a distance of 10m, and Class 3 devices achieve a distance of 1m. The vast majority of Bluetooth enabled devices fall into the Class 2 category, which is ideal for short range wireless communication [5]. Huang and Rudolph [5] suggest that the distances achievable by the various Bluetooth power classes are merely guidelines, and that the actual distances achieved are heavily influenced by surrounding obstacles and other transmissions within the 2.4 GHz frequency band.

Apart from the various power classes of Bluetooth which impose limits on the maximum transmission distance between two devices, another limiting factor which severely constrains the Bluetooth protocol and its suitability for scalable applications is the number of simultaneous connections in Personal Area Networks (PAN). Although Bluetooth was originally intended as a cable replacement for Point to Point connections, it fast became popular due to its ability to create PANs which are otherwise known as ad-hoc networks [6]. PANs enabled multiple devices (up to eight) to communicate with one another through a Master node [7].

Bluetooth Piconets and Scatternets are synonymous with PANs. When there is one Master node and one Slave node a Point to Point connection is established and a Piconet is formed. A Piconet is also formed when there are multiple Slave nodes which are connected to one Master node (Point to Multipoint). In both cases the Slave nodes follow the frequency hopping sequence of the Master node [7].

The obvious limitation with Bluetooth PANs is the maximum of eight active devices in the same Piconet. Scatternets, although not clearly defined by the Bluetooth Special Interest Group (SIG), are formed when multiple Piconets communicate with one another. In Scatternets, a device can be a Master node in one Piconet and Slave node in another Piconet; or a Slave node in both Piconets; but never a Master node in both Piconets [7].

Bluetooth bandwidth is yet another very important consideration where scalability is concerned. According to Huang and Rudolph [5] the theoretical asymmetric data rate achieved between two Bluetooth devices is 723.2 kilobits per second (kbps) and the maximum theoretical symmetric data rate is 433.9 kbps. Asymmetric transmission in this case refers to one Bluetooth device transmitting while the other one receives the transmission. Symmetric transmission occurs when both devices are transmitting and receiving. Sahd [8] found the average asymmetric data rate between a Nokia N95 8GB and Nokia N82 to be 136.17 Kilobytes per second (KBps), which is slightly faster than the theoretical transfer speeds suggested by Huang and Rudolph [5]. Now that we have provided an overview of the Bluetooth protocol and it's constraints, we are able to more efficiently design applications which take all of these constraints into account and make the necessary adjustments. The following sections will motivate why Bluetooth is by no means an outdated protocol, and how it can be used in combination with 802.11 wireless for the implementation of a low cost community telephone network. The next section provides an overview of the 802.11 WLAN protocol.

B. *802.11 WLAN Overview*

According to Gast [9], the two main benefits of wireless networks are mobility and flexibility. With the world in which we live, becoming increasingly more mobile, the need for mobile technologies and the various forms in which they occur (flexibility) also increases. Products based on the 802.11 protocol were initially released in 1997, with the 802.11 consisting of three main layers: The Infrared (IR) layer, and two spread-spectrum radio layers: frequency hopping (FH) and direct sequence (DS).

There are four widely used 802.11 standards: 802.11a, 802.11b, 802.11g, and 802.11n. Each of these standards barring 802.11n (completely new multi-streaming standard) is either an improvement or modification of one of the other standards. Even though these standards are collectively known as the 802.11 standards, they are often referred to as 802.11 wireless.

There are two main frequency bands in which these standards operate: the 2.4 GHz *unregulated* frequency band, and the 5.0 GHz *regulated* frequency band. 802.11a operates in the 5GHz frequency band, and achieves a maximum throughput of 54 Mbps. 802.11b operates in the 2.4 GHz frequency band (same band as Bluetooth) and achieves a maximum throughput of 11 Mbps. 802.11g operates in the 2.4 GHz frequency band and has a maximum throughput of 54 Mbps. 802.11n can operate in the 2.4 GHz or 5.0 GHz frequency bands, or both simultaneously. 802.11n achieves a maximum throughput of 600 Mbps due to the use of wider channels (40 MHz instead of 20 MHz). With that said, 802.11n is backward compatible with 802.11a, 802.11b, and 802.11g, thus making it extremely versatile and by far the best choice for scalable applications which require large amounts of bandwidth. Since the 2.4 GHz frequency band is unregulated, there is a lot more interference which occurs in this band, and as a result the bandwidth measurements vary according to the amount of interference encountered. When the 802.11n standard utilizes the 5.0 GHz frequency band, the number of overlapping channels increases due to the wider channel width of 40 MHz.

One of the common questions asked is whether Bluetooth and 2.4 GHz based 802.11 wireless encounter issues with interference when in close proximity to one another. Luckily both protocols have measures to deal with interference in the unlicensed 2.4 GHz frequency band. In order to deal with interference, Bluetooth employs a frequency hopping scheme, which enables it to transmit each packet on a different frequency. Bluetooth is generally comprised of 79 channels (dependent on country of operation), and the transmission time allowed for each slot is 625μs, thus allowing Bluetooth to achieve 1600 hops per second, which significantly reduces the likelihood of interference. The frequency hopping scheme is determined by the master node in the Piconet. With that said it is clear that Bluetooth employs a very sophisticated interference avoidance model in the unlicensed and often interference riddled 2.4 GHz frequency band.

The 2.4 GHz frequency band (on which 802.11b and 802.11g operate) is comprised of fourteen channels with each channel separated by 5 MHz. The United States only has 11 channels available in the 2.4 GHz frequency band due to policies enforced by the Federal Communications Commission (FCC). 802.11b and 802.11g wireless require a 22 MHz range to modulate the wireless signal, and since each channel in the 2.4 GHz frequency band is separated by 5 MHz, there is a substantial amount of channel overlap [10].

With the large amount of channel overlap in the 2.4 GHz frequency band, there are only three non-overlapping channels: Channel 1; Channel 6; and Channel 11. What this effectively means, is that even though wireless devices (on the 2.4 GHz frequency band) are on different channels, they still interfere with one another apart from the aforementioned channels.

Unlike Bluetooth, 802.11b and 802.11a operate on a fixed channel which makes them a lot more susceptible to interference.

The next section introduces the Android platform and the operation of Bluetooth and 802.11 wireless on this platform.

III. ANDROID

Android is often described as a software stack for mobile devices, and is comprised of an operating system, middleware, and a plethora of applications [12]. Android's architecture is comprised of a hierarchical stack of components, with applications forming the top most layer, and the Linux kernel forming the lowest layer [13]. Figure 1 depicts the Android architecture or component stack:

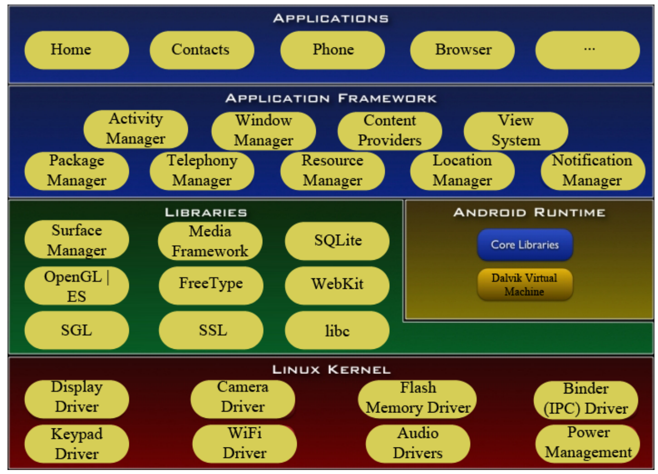

Figure 1. Android component stack

From Figure 1 it can be seen that Android's lowest layer is the Linux kernel upon which all the other layers run. Android Libraries and runtime are situated above and are dependent on the Linux kernel. The Application framework provides the necessary services and systems which applications are dependent upon. Android comes standard with a set of core applications (email, SMS, calendar, etc.) and other applications written in the Java programming language can be installed [12].

One of the most important components of the Android component stack, which is located in the Android Runtime, is the Dalvik Virtual Machine [14]. The Dalvik Virtual Machine is Android's answer to the Java Virtual Machine (JVM), and it enables multiple instances of the Virtual Machine to be run simultaneously and takes advantage of the Linux operating system for process isolation and security [15]. Dalvik Virtual Machine differs from the JVM in that it executes .dex files which are converted from .class and .jar Java files at compile time. The benefit of the .dex files is that they are more compact and efficient than their Java equivalents and this proves to be a very important consideration for devices with limited battery and memory [15].

According to Meier [14], Android belongs to a new wave of mobile operating systems which have been designed for the increasingly more powerful hardware. With the increased power of the hardware of mobile devices, comes an increase in the capability and scalability of services which can be hosted on these devices. Meier [14] describes a number of application services which are crucial in providing developers with the necessary functionality required for

 ISBN: 978-1-61208-210-3

application development: Activity Manager; Views; Notification Manager; Content Providers; and the Resource Manager. The Activity Manager controls the lifecycle and management of activities; Views are a means of constructing the interfaces for your activities; The Notification Manager provides a means for signaling users with notifications related to the activities described above; Content Providers enable inter application communication through the sharing of data; The Resource Manager supports non-code resources like strings and graphics.

With an introduction to the Android platform, the next section introduces the concept of Bluetooth proxying and describes the proposed infrastructure.

IV. BLUETOOH PROXYING AND INFRASTRUCTURE

Before we look at Bluetooth proxies and how they operate, it is necessary for a definition of proxy servers themselves. Indiana University [16] defines a proxy server (or application level gateway) as a computer that acts as a gateway or intermediary between one network and another. For the purposes of this research and as a general rule of thumb, proxy servers need not be implemented on a computer alone. Proxy servers are generally found at the Network layer (Layer 3) or higher of the Open Systems Interconnection (OSI) model [22], and they provide vital services such as the forwarding of traffic between source and destination, as well as efficiency improvements through information retrieval through proxy cache's [16]. Figure 2 depicts the various layers in the OSI model:

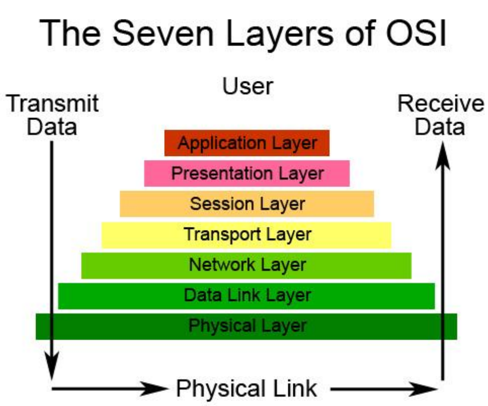

Figure 2. OSI Network Model [17]

Depending on which OSI layer proxy servers operate on, their implementation is somewhat different. Android provides a wide variety of libraries which enable seamless access to lower level resources from layers high up in the OSI model (presentation and application layers). As such, we propose the implementation of an application level proxy which enables the seamless communication between devices on a Bluetooth network and those on an 802.11 wireless network.

A. *Bridging and communication*

Theoretically, it would be possible to bridge the Bluetooth and 802.11 protocols if they were operating on the same channel and if necessary adjustments were made to both protocols to enable them to communicate with one another. Another look at Section 2, Subsection B would show that due to the interference avoidance models of the Bluetooth protocol, it would be highly unlikely that an approach to enable communication between the two protocols would be implemented at OSI layers below the Presentation layer (layer 6).

In order for the two protocols to communicate with one another, they need a means of communication. Since there is no existing method for enabling this communication between Bluetooth and 802.11 wireless, we propose an application level framework called *Blue Bridge*, which essentially intercepts and interprets information from one source and forwards it to another. *Blue Bridge* is built on the Android platform, and is ideally run on a high end Android phone such as the Samsung Galaxy SII [18]. The Samsung Galaxy SII referred to as the "Samsung" from now on is comprised of a 1.2 GHz Cortex-A9 processor; 802.11 a/b/g/n wireless; Bluetooth V3.0+HS; 1 GB RAM and 16 GB internal storage. The Samsung runs V2.3.4 Android (Gingerbread) and is upgradable is V4.x [19]. With high end specifications such as these, the Samsung should be more than capable of intercepting connections on either the Bluetooth or 802.11 interfaces.

The Android platform was chosen as the primary platform for *Blue Bridge* due to its wide scale adoption in high end cell phones (which are necessary for the purposes of this research), and its ability to establish Bluetooth and 802.11 wireless connections at the application level. The Samsung was chosen was the device of choice for *Blue Bridge* over and above conventional wireless routers, due to the Bluetooth and 802.11 wireless interfaces already being present in the device. The Samsung was also chosen due to its size, low power requirements, and the fact that this device could be concealed and kept safe from prying eyes. Conventional 802.11 wireless routers are rarely equipped with Bluetooth interfaces and as such Android enabled cellphones were chosen as a more appropriate platform which tackles the issue at hand.

Blue Bridge creates a Bluetooth PAN and an 802.11 wireless hotspot which enables nodes from either network to connect to the Samsung and essentially proxy communications through the *Blue Bridge* node to one another. For the purposes of simplicity we will refer to the Bluetooth interface as Interface 1, and the 802.11 wireless interface as Interface 2. There are two methods which we propose for the receiving and forwarding of data from one interface to another:

1. Any data received on Interface 1 of the Samsung *Blue Bridge* receives all of the data before forwarding / retransmitting it to the destination device on Interface 2.

 ISBN: 978-1-61208-210-3

2. Any data received on Interface 1 of the Samsung *Blue Bridge* forwards / retransmits data on the fly from the streams and buffers handling the current transmission.

On a powerful Android device such as the Samsung, Method 1 is feasible in theory, but obvious limitations exist, such as the scenario where large files are being received from Interface 1, and the entire file has to be fully received before retransmission occurs, which leads to time delays and heavy loads being placed on the Samsung. Another limitation of Method 1 is the fact that there has to be sufficient flash memory capacity on the Samsung before the initial transmission of the file. In the event that there is insufficient flash memory capacity, mechanisms would have to be implemented to inform the client node of oversized files, which further increases the total transmission time to the destination node. The benefit of using this method is that files can be checked for integrity before retransmission and the chances of file corruption are minimized.

Method 2 is optimized for efficiency and retransmits data to the destination device as soon as it is received, which results in a significantly faster transfer time than if the file were to be received in its entirety as in Method 1. This method is also without the problem of the received file size exceeding the capacity of the Samsung. This is due to the fact that parts of the file are received, temporarily stored in buffers and then immediately transmitted from these buffers to the destination device. This Method cannot check files for integrity as with Method 1, due to the fact that only parts of the file are received and then immediately transmitted and forgotten about.

Regardless of whether Method 1 or Method 2 is used for *Blue Bridge*, it would be considered wasteful for a Bluetooth device to communicate with another Bluetooth device in range by forwarding all traffic through *Blue Bridge*, instead of establishing direct connections between one another. The primary function of *Blue Bridge* is to forward traffic from one interface to the other, and not to retransmit data on the same interface it was received on. Apart from application level receiving and forwarding, there needs to be a method by which *Blue Bridge* can inform nodes on both the Bluetooth and 802.11 wireless networks of one another. Subsection B proposes a very lightweight protocol, *Linkage*, which informs nodes of one another and controls the transmission and receiving between a node and the *Blue Bridge*.

B. Communication between nodes and Blue Bridge through Linkage

As mentioned above, the primary function of *Linkage* is to enable communication between nodes in one network and nodes in another, by making use of the central *Blue Bridge*. The most obvious approach to the implementation of *Linkage* is by programming the *Blue Bridge* node to broadcast packets to all nodes on either interface informing them of every other node. Broadcasting packets on a network is inefficient, and can consume the minimal bandwidth available in protocols such as Bluetooth. The way in which a broadcast protocol would work in this instance is for the *Blue Bridge* to broadcast a list of nodes within its range to every other node on the network. The *Blue Bridge* would have to periodically scan the network to determine which nodes are still in range, and then broadcast this information which could result in large amounts of network traffic where multiple *Blue Bridges* are concerned.

Although cellular networks in South Africa are expensive, the model which they employ with each cell phone having a number, is an ideal system to avoid broadcast traffic. Instead of having to maintain a database of nodes within range, we propose a system whereby nodes get assigned a number upon initial contact with the *Blue Bridge*. By employing this system, nodes can contact each other through the *Blue Bridge* by using a number which remains constant, thus eliminating the need for constant maintenance of nodes in range by the *Blue Bridge*. Figure 3 shows how nodes communicate using a number based system with *Linkage:*

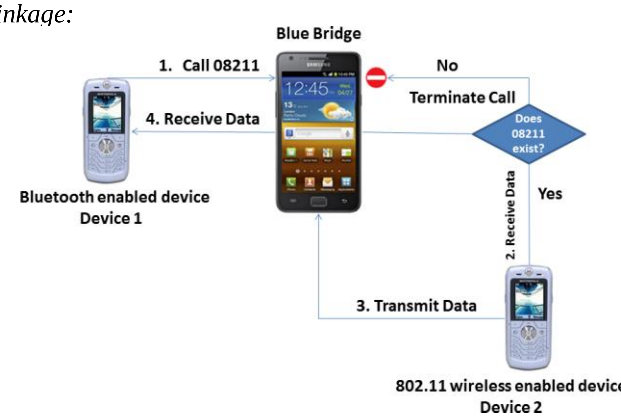

Figure 3. Linkage controlling a call between two phones

From Figure 3 it can be seen that Device 1 tries to establish communication with Device 2 by communicating through the *Blue* Bridge. Device 1 has its Bluetooth interface enabled and Device 2 has its 802.11 wireless interface enabled, which means that without the *Blue Bridge* communication between the two devices cannot take place. After the initial communication from Device 1, the *Blue Bridge* finds the associated address (either a Bluetooth or 802.11 address) and attempts to establish a connection. If Device 2 is unreachable after a certain timeout period, the call is terminated. If the *Blue Bridge* is able to reach Device 2, a connection is established and the data is transmitted from Device 1 through the *Blue Bridge* to Device 2. Of course if the type of connection were a voice call, Device 2 would have to transmit data to Device 1 through the *Blue Bridge* as well. The most crucial component to enabling communication between Bluetooth and 802.11 wireless devices is the allocation of a permanent number to each device from either network, which associates with the *Blue Bridge*. The assignment of numbers to devices is beyond the scope of this paper.

Figure 3 depicts how one 802.11 wireless device communicates with one Bluetooth device. The maximum number of active Bluetooth connections in one Piconet is 7, which limits the number of phone calls which can occur simultaneously through the *Blue Bridge*. Figure 5 shows an SDL diagram depicting the processes involved in handling simultaneous connections and ability of *Blue Bridge* to forward connections from one interface to another to reach the destination device. From Figure 5, it can be seen that *Blue Bridge* begins the bridging process by waiting for new connections. When a new connection is established, *Blue Bridge* determines whether the destination device is reachable. If the destination device either does not exist or is unreachable, the connection is terminated thus freeing up resources. If the destination device is found, the protocol with which the destination device and the *Blue Bridge* are communicating is determined. If the communicating protocol is 802.11 wireless, data is forwarded from the source device, through the *Blue Bridge*, to the destination device by implementing 802.11 wireless communication between the *Blue Bridge* and the destination device. If the communicating protocol is Bluetooth, the number of available connections is determined. This is necessary, since a maximum of 7 active Bluetooth connections can exist in the same Piconet. If the number of active connections is less than or equal to 7, then data is forwarded between from the source device, through the *Blue Bridge*, to the destination device by implementing Bluetooth communication between the *Blue Bridge* and destination devices. The above process is repeated until the concurrent Bluetooth connections is greater than 7, at which point 802.11 wireless communication will serve as the only means of communication for additional connections established.

With a proposed model for enabling Bluetooth and 802.11 communication highlighted in this section, the next section proposes a model built on top of *Blue Bridge* which aims to expand the reach of wireless networks.

V. NETWORK EXPANSION THROUGH SSID AND CHANNEL MATCHING AS A POSSIBLE EXPANSION

Although the primary focus of this paper is the proposal of an architecture which enables communication between the Bluetooth and 802.11 wireless protocols, this section proposes a model which describes the expansion of 802.11 wireless networks through *Blue Bridge* nodes, and hence the interconnection of these nodes, thus creating a community telephone network. According to [20] and [21], there are three prerequisites for the expansion of wireless networks: Identical SSIDs; Identical wireless encryption; and ideally the same wireless channel.

Android provides the ability to create wireless hotspots on a particular channel, with or without security (wireless encryption) enabled. With such functionality available, an existing wireless network could be expanded by creating a hotspot with the same SSID, on the same channel, with the same encryption. The *Blue Bridge* could enable inter-protocol communication as well as the expansion of wireless networks. The same principle could be applied to 802.11 wireless nodes running the Android platform, thus creating a wireless mesh network with minimal infrastructure.

Rural areas in South Africa are generally within range of nearby towns and cities, which allows for long distance 802.11 wireless links (50km) to bridge remote community telephone networks. Although there is no shortage of 802.11 wireless networks in South Africa, the idea of making use of existing networks for the purposes of this research would not be feasible due to a plethora of factors relating to network type (802.11a, 802.11b, 802.11g, and 802.11n) and the associated bottlenecks; open ports; and network names and channels. With that said, we have opted to implement our own low cost infrastructure to ensure availability and scalability.

In order to connect multiple *Blue Bridge* nodes across large geographical areas, directional antennas are proposed. Each *Blue Bridge* will connect to the directional antenna using the 802.11n wireless protocol. Figure 4 shows the interconnection of *Blue Bridge* via directional antennas:

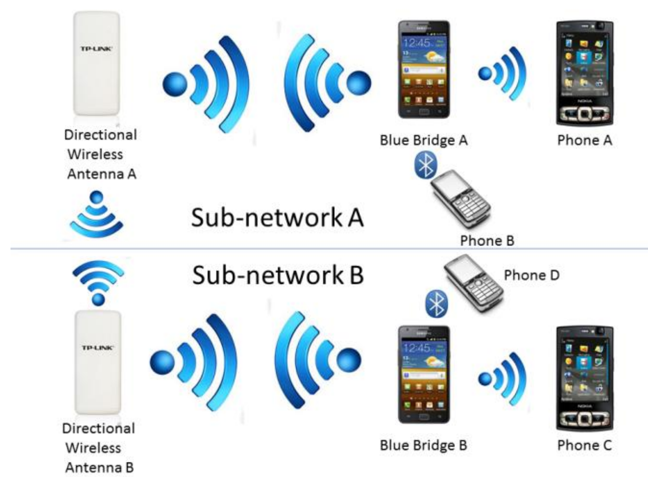

Figure 4. Interconnection of Blue Bridges

Figure 4 depicts two networks, Sub-network A, and Sub-network B, both of which are comprised of a directional wireless antenna, the *Blue Bridge*, and two cell phones. Sub-networks A and B enable communication between nodes in either network via the directional wireless link between Antennas A and B. *Blue Bridge A* enables access to the community telephone network via its 802.11 wireless and Bluetooth interfaces, thus enabling communication with Phone A and Phone B using the respective interfaces. Similarly, *Blue Bridge B* communicates with Phone C via the 802.11 wireless interface and with Phone D via the Bluetooth interface.

Although Sub-network A and Sub-network B can communicate via the wireless link between Antenna's A and B, this does not enable automatic communication between *Blue Bridge A* and *Blue Bridge B.* In order to facilitate the communication between the *Blue Bridges*, we propose the use of a dedicated registration server which keeps track of IP address assignments to each *Blue Bridge*. The registration server also handles the assignment of IP addresses to 802.11 wireless and Bluetooth based nodes connecting to the *Blue Bridge.* Since this paper focuses on bridging the 802.11 wireless and Bluetooth protocols, the operation and subsequent assignment of addresses to *Blue Bridges* and other nodes is beyond the scope of this paper.

This section provided a brief overview of 802.11 wireless network expansion and showed yet another powerful aspect of the Android platform. This section also showed how high powered directional wireless antennas can be used to link multiple *Blue Bridges*. The next section concludes this paper.

VI. Conclusion

Although protocols such as Bluetooth are being used less in first world nations where the majority of mobile handsets possess 802.11 wireless capability, Bluetooth is a protocol, which is still widely employed in the African context. Bluetooth, although limited in terms of scalability, can serve as a point of access to community telephone networks for people with mobile handsets which lack 802.11 wireless capability. This paper showed how existing protocols can be combined through application level bridging on the *Blue Bridge* and the employment of the proposed protocol, *Linkage*. This paper highlighted the benefits and constraints of multiple methods of implementation and illustrated which method was chosen and why. This paper also proposed a future extension to the *Blue Bridge* which extends the range of wireless networks through SSID, encryption, and channel matching.

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. Last but not least, I would sincerely like to thank Rhodes University Computer Science Department of their constant support and invaluable resources.

References

[1] Hoovers. Bluetooth on the road. Available at: http://www.hoovers.com/business-information/--pageid__13751--/global-hoov-index, 2012. [Accessed 08-03-2012].

[2] Geier, J. Is Bluetooth still a threat to 802.11?. Available at: http://www.wi-fiplanet.com/tutorials/article.php/2210461, 2003. [Accessed 14-04-2012].

[3] GSMARENA. Bluetooth 3.0 specs revealed, goes up to 11 thanks to 802.11. Available at: http://www.gsmarena.com/bluetooth_30_specs_revealed_goes_up_to_11_thanks_to_80211-news-885.php, 2009. [Accessed 14-04-2012].

[4] NOKIA. Bluetooth technology overview. Available at: www.forum.nokia.com, April 2003. [Accessed 07-04-2011].

[5] Huang, A. and Rudolph, L., "Bluetooth essentials for programmers," 2007. Cambridge University Press.

[6] Mackie, D.S., "Extending the reach of personal area networks by transporting Bluetooth communications over IP networks," 2007. Rhodes University.

[7] Mander, J. and Picopoulos, D., "Bluetooth Piconet applications."

[8] Sahd, C., "Bluetooth audio and video streaming on the the J2ME platform," 2010. Rhodes University.

[9] Gast, M., "802.11 wireless networks: the definitive guide," 2005. O'Reilly Media.

[10] Balchunas, A. Introduction to 802.11 wireless. Available at: www.routeralley.com, 2010. [Accessed 19-04-2012].

[11] Fainberg, M., "A performance analysis of the ieee 802.11 b local area network in the presence of bluetooth personal area network," 2001. Polytechnic University.

[12] Android Developers. What is Android?. Available at: http://developer.android.com/guide/basics/what-is-android.html, 2012. [Accessed 28-03-2012].

[13] Linux for you. A developer's first look at Android. Available at: http://www.linuxforu.com, 2008. [Accessed 19-04-2012].

[14] Meier, R., "Professional Android 2 application development," 2012. Wrox.

[15] Burnette, E., "Hello, Android: introducing Google's mobile development platform," 2009. The Pragmatic Bookshelf.

[16] Indiana University. What is a proxy server?. Available at: http://kb.iu.edu/data/ahoo.html, 2011. [Accessed 15-04-2012].

[17] University of Washington. The OSI Model. Available at: http://www.washington.edu/lst/help/computing_fundamentals/networking/osi, 2012. [Accessed 15-04-2012].

[18] Samsung. Samsung Galaxy SII. Available at: http://www.samsung.com/global/microsite/galaxys2/html/, 2012. [Accessed 05-04-2012].

[19] GSMARENA. Samsung I9100 Galaxy SII. Available at: http://www.gsmarena.com/samsung_i9100_galaxy_s_ii-3621.php, 2012. [Accessed 06-04-2012].

[20] DD-WRT. Wlan repeater. Available at: http://www.dd-wrt.com/wiki/index.php/Wlan_Repeater, 2012. [Accessed 06-04-2012].

[21] Geier, J., "Extending WLAN range with repeaters," 2004. Available at: http://www.wi-fiplanet.com/tutorials/article.php/1571601. [Accessed 10-04-2012].

[22] Zimmerman, H. "OSI Reference Model – The ISO model of architecture for open systems interconnection. IEEE Transactions on Communications," 1980.

[23] B. P. Crow, I. K. Widjaja, G. Jeong, and P. T. Sakai. "IEEE-802.11 Wireless Local Area Networks," IEEE Communications Magazine, pp. 116-126, Sept. 1997.

 ISBN: 978-1-61208-210-3

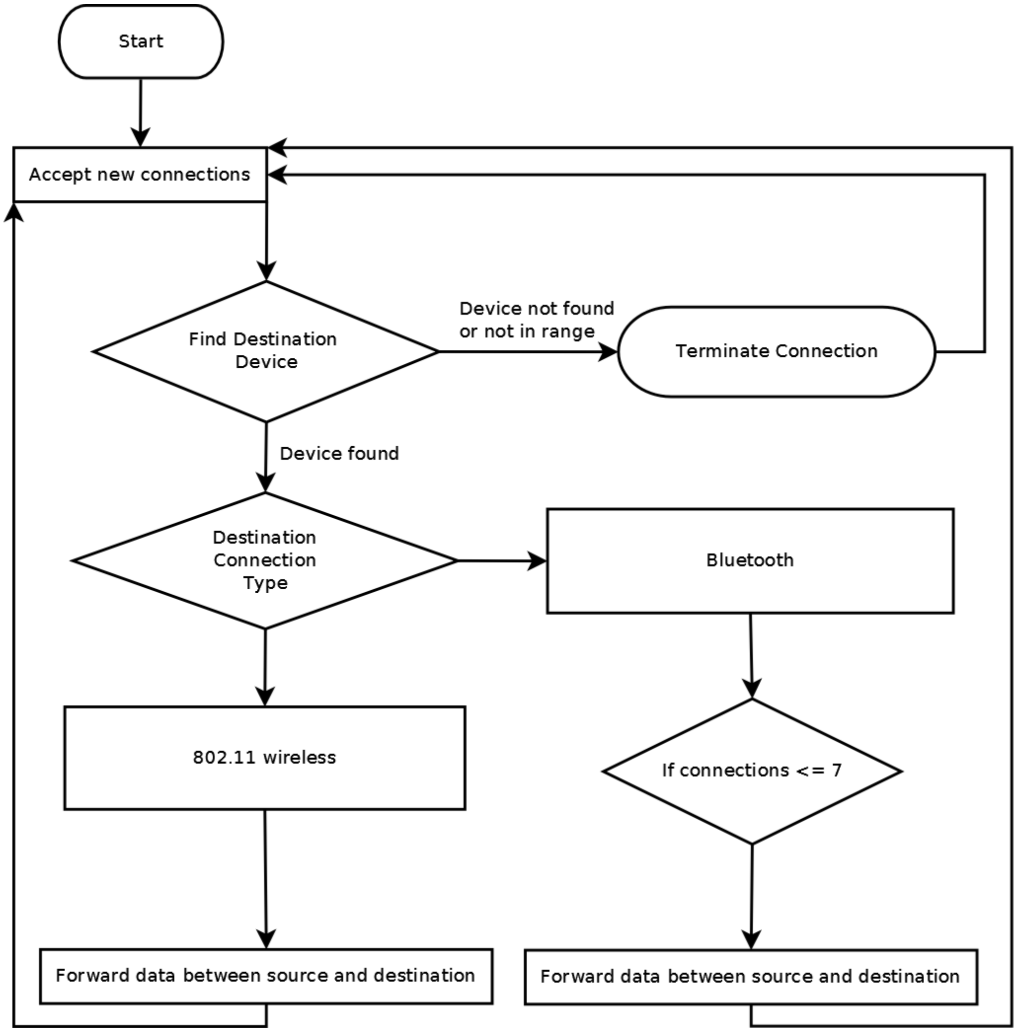

Figure 5. Blue Bridge processes and procedures

 ISBN: 978-1-61208-210-3