

# ENERGY CONSERVATION IN HETEROGENEOUS SMARTPHONE AD HOC NETWORKS

By

James Mariani

A THESIS

Submitted to  
Michigan State University  
in partial fulfillment of the requirements  
for the degree of

Computer Science – Master of Science

2018

## **ABSTRACT**

### **ENERGY CONSERVATION IN HETEROGENEOUS SMARTPHONE AD HOC NETWORKS**

By

James Mariani

In recent years mobile computing has been rapidly expanding to the point that there are now more devices than there are people. While once it was common for every household to have one PC, it is now common for every person to have a mobile device. With the increased use of smartphone devices, there has also been an increase in the need for mobile ad hoc networks, in which phones connect directly to each other without the need for an intermediate router. Most modern smart phones are equipped with both Bluetooth and Wifi Direct, where Wifi Direct has a better transmission range and rate and Bluetooth is more energy efficient. However only one or the other is used in a smartphone ad hoc network. We propose a Heterogeneous Smartphone Ad Hoc Network, HSNet, a framework to enable the automatic switching between Wifi Direct and Bluetooth to emphasize minimizing energy consumption while still maintaining an efficient network. We develop an application to evaluate the HSNet framework which shows significant energy savings when utilizing our switching algorithm to send messages by a less energy intensive technology in situations where energy conservation is desired. We discuss additional features of HSNet such as load balancing to help increase the lifetime of the network by more evenly distributing slave nodes among connected master nodes. Finally, we show that the throughput of our system is not affected due to technology switching for most scenarios. Future work of this project includes exploring energy efficient routing as well as simulation/scale testing for larger and more diverse smartphone ad hoc networks.

## **ACKNOWLEDGEMENTS**

I would first like to thank my advisor, Dr. Li Xiao, who was instrumental in helping and guiding me throughout my years in graduate school as well as this thesis. Without her help I would not have been able to complete this thesis successfully. I would also like to thank my review committee, Dr. Esfahanian and Dr. Kulkarni for their support throughout this process.

Secondly, I would like to thank my parents for their support and encouragement throughout my studies.

I would like to thank my friend and colleague Spencer Ottarson. This project started as a joint venture and although things have changed a lot from the beginning, his help and work in the early days were instrumental in making this project a reality.

I would also like to thank my teaching advisor Dr. Dyksen for his guidance and advice in all matters of my life. He has helped me and supported me throughout the years and I appreciate everything.

Lastly, I would like to thank all the faculty and staff in the Michigan State University department of Computer Science. When I arrived here as an undergrad 6 years ago I had no idea what I wanted to do with my life or what my passion was, but throughout the years they have helped me to realize what I am capable of and what I plan to do with my life.

## TABLE OF CONTENTS

LIST OF TABLES . . . . .	vi
LIST OF FIGURES . . . . .	vii
CHAPTER 1 INTRODUCTION . . . . .	1
1.1 Motivation . . . . .	1
1.2 Challenges . . . . .	3
1.3 Overview of Heterogeneous Smartphone Ad Hoc Network . . . . .	3
1.4 Organization . . . . .	4
CHAPTER 2 BACKGROUND INFORMATION . . . . .	5
2.1 Bluetooth . . . . .	5
2.2 Wifi Direct . . . . .	5
2.3 Performance Differences . . . . .	7
CHAPTER 3 RELATED WORK . . . . .	8
3.1 Battery Consumption in Smartphones . . . . .	8
3.2 Bluetooth or Wifi Direct Ad Hoc . . . . .	8
3.3 Heterogeneous Ad Hoc Networks . . . . .	9
3.4 Switching without Infrastructure . . . . .	10
3.5 Switching with Infrastructure . . . . .	11
3.6 Cross Technology Communication (CTC) . . . . .	11
CHAPTER 4 DESIGN OF HETEROGENEOUS SMARTPHONE AD HOC NETWORK .	12
4.1 Design Overview . . . . .	12
4.2 Technical Details . . . . .	13
4.2.1 Modified Flooding . . . . .	13
4.2.2 Mode Switching . . . . .	14
4.2.3 Master Node Load Balancing . . . . .	16
4.3 HSNet Chat Application . . . . .	19
4.4 Network Topology . . . . .	21
CHAPTER 5 PERFORMANCE EVALUATION . . . . .	24
5.1 Evaluation Methodology . . . . .	24
5.1.1 Testbed . . . . .	24
5.1.2 Evaluation Metrics . . . . .	25
5.2 Slave Node Analysis . . . . .	28
5.3 Master Node Analysis . . . . .	29
5.4 Throughput Analysis . . . . .	34
CHAPTER 6 CONCLUSIONS AND FUTURE WORK . . . . .	41
6.1 Conclusion . . . . .	41

6.2	Future Work . . . . .	42
6.2.1	Routing . . . . .	42
6.2.2	Simulation/Scale . . . . .	42
BIBLIOGRAPHY . . . . .		44

## LIST OF TABLES

Table 2.1: Comparison of Bluetooth and WiFi Direct performance . . . . .	7
Table 5.1: Energy consumption and lifetime analysis of slave nodes . . . . .	29
Table 5.2: Energy consumption of master nodes before and after load balancing . . . . .	33

## LIST OF FIGURES

Figure 2.1: Connected piconets . . . . .	6
Figure 4.1: Example mode switching scenarios . . . . .	15
Figure 4.2: Example of an unbalanced network topology . . . . .	16
Figure 4.3: Load balancing of master nodes . . . . .	18
Figure 4.4: Menu showing the different options and settings of our system . . . . .	20
Figure 4.5: Screenshot of our application that showcases our autosending feature, the switching of technologies, and the options menu . . . . .	22
Figure 4.6: Example network topology . . . . .	23
Figure 5.1: The results of our experiments regarding energy consumption of slave nodes. The top graph is a slave node sending by only Bluetooth. The middle graph shows a slave node sending by only Wifi Direct. The Bottom graph shows a slave node that uses our technology switching paradigm for battery savings . . .	27
Figure 5.2: Power consumption of a Bluetooth master node with respect to the number of slaves . . . . .	30
Figure 5.3: Power consumption of a Wifi Direct group owner with respect to the number of group members . . . . .	31
Figure 5.4: Results of an experiment to showcase the load balancing system for master nodes	32
Figure 5.5: Throughput analysis of communication within a single piconet . . . . .	35
Figure 5.6: Throughput analysis of communication between two piconets . . . . .	37
Figure 5.7: Throughput analysis with respect to the threshold for switching technologies . .	40

# CHAPTER 1

## INTRODUCTIONS

This chapter will introduce this thesis, including motivation for the project, research challenges, and an overview of the HSNet system.

### 1.1 Motivation

Due to a recent push towards mobile devices and mobile computing, many new fields and concepts have emerged in networking. One issue that arises, however, is that when we are so reliant on mobile devices for communication, if the centralized network goes down it leaves many people unable to communicate. We have seen examples of this occur during natural disasters, such as Hurricane Sandy and the East Japan Earthquake [1]. Because of this, the need and interest in decentralized networks has never been higher. One of the more promising and interesting fields regarding mobile devices and decentralized networks is the idea of a proximity service (ProSe) application [2]. A ProSe application is an app that tries to find instances of the same app running on devices physically near to itself. Once a ProSe application finds someone close to itself, the end result is the exchange of information [2].

This makes ProSe applications unique from traditional applications in that they are “location centric” and can only communicate with others near the device instead of a typical “friends list” type application [2]. There are two major use cases for ProSe applications: public safety communications and discovery mode applications [3]. Public safety communication applications can be used in case of a network outage due to any number of things including inclement weather or your device being out of range. Discovery mode applications include things such as mobile social networking apps [3].

There are two main categories of ProSe applications: Over-the-top (OTT) and device-to-device (D2D) [4]. OTT applications rely on a server with periodic location updates from all users. This introduces a large amount of overhead that is not energy efficient. The topic of this thesis will



focus specifically on D2D ProSe applications. D2D solutions are developed so that two devices can communicate directly with each other without using an existing network. In this type of application you are restricted in who you can communicate with by the signal strength of the technology being used. The two major technologies used to implement D2D communications are Bluetooth and WiFi Direct [2].

Within the realm of D2D applications, there are two different (but similar) types of applications. There are single-hop and multi-hop ad hoc networks. A single-hop ad hoc application creates a network in which it is only possible to communicate with a node within your device's current range. In a single-hop network each individual node does not act as a router and will not forward on information to its neighbors [5]. This work, however, is focused on the second type of D2D network, which is a multi-hop ad hoc network. A multi-hop ad hoc application creates a network in which each individual node can act as a router and forward on information and messages it receives to its neighbors [5]. This greatly expands the reach of the network and communications.

In almost all types of ad hoc networks, one major hurdle is the battery power of individual nodes. When working with ProSe and mobile/smartphone ad hoc networks this concern is heightened due to the fact that users will use their phone for things other than communication, which will drain the battery even faster. In the case of a natural disaster where communication is of the utmost importance, if the nodes that make up the ad hoc network run out of battery then you run the risk of disconnecting large chunks of the population.

To address these issues, we propose a heterogeneous smartphone ad hoc network, HSNet, a multi-hop ad hoc network in which devices can communicate with each other even if they are not using the same technology (Bluetooth or WiFi Direct). In addition to connecting devices running multiple technologies we employ a mode switching algorithm that will conserve energy on individual nodes by adaptively switching between communication technologies to prolong battery life and maintain high throughput.

## 1.2 Challenges

In this section we discuss some of the challenges that come with developing smartphone ad hoc networks as well as challenges with conserving energy.

Traditionally, D2D applications use one of Bluetooth or WiFi Direct to facilitate communication between devices. WiFi Direct is a more powerful ad hoc communication technology and boasts greater range and throughput than Bluetooth, but this comes with the cost of greater energy consumption. Bluetooth remains a popular communication technology for ad hoc networks due to its ubiquity and its energy efficiency. Ad hoc networks utilizing only one communication technology runs into a few problems regarding devices being left out if they are not compliant with one of the technologies, as well as energy concerns if you are using WiFi Direct.

When working with smartphones we run into another problem with only being able to make changes in the application layer. This compounds the problem of trying to make a heterogeneous network that uses multiple technologies. Because we can only make changes to the application layer we do not have as much control over the network stack and have to rely on publicly available APIs that are not designed for constant technology switching.

Developing for smartphones in general places a large emphasis on energy consumption. Smartphones forming an ad hoc network compounds this concern as our framework has to be able to allow the phone to function normally while still maintaining the connection to the network. We do not have control over what else the user might do with their phone, and our framework has to be able to handle this uncertainty by remaining as energy efficient as possible.

## 1.3 Overview of Heterogeneous Smartphone Ad Hoc Network

HSNet allows devices running different technologies to communicate seamlessly, while offering energy savings to each device, through a *mode switching* algorithm to automatically decide what technology to use, as well as master node *load balancing*, all while maintaining maximum throughput of the network. To evaluate these goals we design a chat application on the Android OS that runs our HSNet platform. We show that we are able to connect devices running different

technologies, and are able to provide power savings on devices running low on battery by switching from WiFi Direct to Bluetooth if available. We also analyze throughput concerns with switching technologies and show how HSNet can maintain high throughput while saving energy.

## **1.4 Organization**

This thesis is organized as follows. Chapter 2 introduces background information regarding Bluetooth, WiFi Direct, as well as their performance differences. Chapter 3 discusses works related this thesis. The design of our proposed solution is discussed in Chapter 4. In Chapter 5 we evaluate the performance of our system, and Chapter 6 concludes this thesis and introduces future work.

## **CHAPTER 2**

### **BACKGROUND INFORMATION**

The proliferation of mobile devices (phones and tablets) has led to an increase in demand for mobile/smartphone networks of connected devices. These networks are called smartphone ad hoc networks, and they are a form of mobile ad hoc networks (MANETs). MANETs are networks of connected mobile devices that communicate and share information device to device without any communication with the internet or outside network. Two of the major technologies used to implement smartphone ad hoc networks are Bluetooth and WiFi Direct. In this section we give a brief overview of the basics of Bluetooth and Wifi Direct, specifically as it pertains to range, throughput, and energy consumption.

#### **2.1 Bluetooth**

Bluetooth and WiFi Direct work in similar ways on mobile devices. Bluetooth employs what it calls a master-slave paradigm in which one master node is connected to up to seven slave nodes. Slave nodes can only communicate with the master, whereas the master node can send to all of its slaves. Therefore a multi-hop system is needed for slaves to communicate with one another. If you assume a network of connected bluetooth devices in which phones A and C are slaves to phone B, in order for A to communicate with C, it has to send through the intermediary phone B which will forward the message on to phone C. Normal Bluetooth devices utilize a sender, receiver, and an antennae to send and receive messages. However, Bluetooth communication in smartphones works by use of a combined sender/receiver, called a transceiver. Both sending and receiving messages goes through the antennae, and therefore sending and received cannot happen simultaneously [6].

#### **2.2 Wifi Direct**

WiFi Direct uses something almost exactly the same, except the term "master" is replaced with "group owner," and "slave" is replaced with "client" or "group member." The only other major

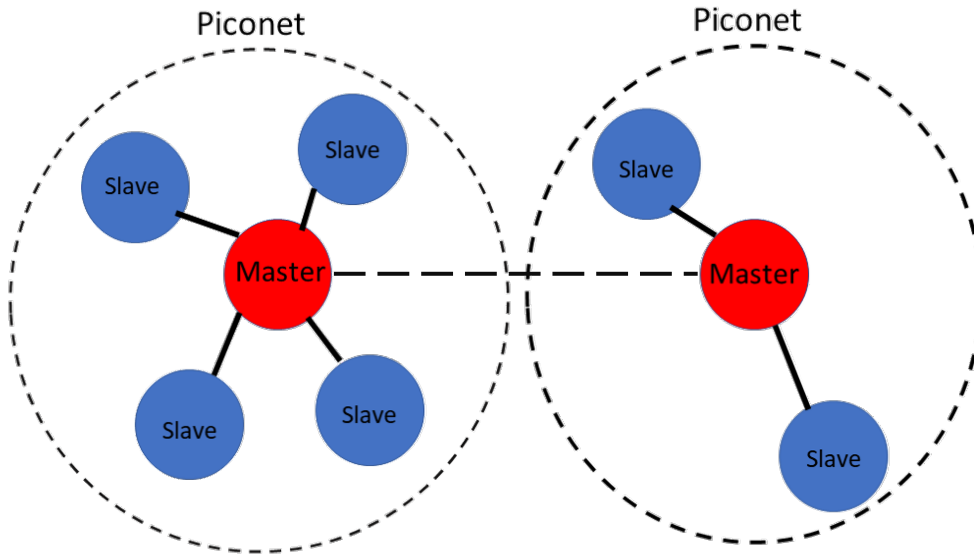


Figure 2.1: Connected piconets

difference at a high level is that a Wifi Direct group consists of one group owner and up to four clients versus seven slaves with Bluetooth. As with Bluetooth, however, in order for two group members to communicate with each other they must first send through the group owner. The same example as given above applies to Wifi Direct. In both Wifi Direct and Bluetooth these networks of master-slave connections are called piconets, and within a network you can have many connected piconets. An example of connected piconets can be seen in 2.1. For the remainder of this work and for clarity, the terms master and group owner will be used interchangeably, as will slave and group member/client.

When creating a Wifi Direct group, there is no way to specify who will be the group owner and who will be the group member. This is based on many factors out of the control of the user, including a negotiation between devices. However, once a group owner is chosen, they act as a DHCP server and then assigns IP addresses to the group members through the DHCP protocol [7].

Table 2.1: Comparison of Bluetooth and WiFi Direct performance

	Range	Transfer Rate	Energy for Sending	Energy for Receiving
Bluetooth	10 m	24 Mbps	520 mW	456 mW
WiFi Direct	60 m	250 Mbps	1560 mW	1448 mW

## 2.3 Performance Differences

In terms of performance, Bluetooth and Wifi Direct have both positives and negatives. In terms of range, Bluetooth has a range of approximately 10 meters [8], whereas Wifi Direct has a range of around 60 meters. Wifi Direct also boasts a transfer rate 250 Mbps, which is more than ten times greater than Bluetooth that has a transfer rate of 24 Mbps [9]. It is clear that WiFi Direct is superior to Bluetooth in terms of throughput and range, but the advantage of Bluetooth comes from its energy efficiency. Sending a Bluetooth message uses about 520 mW, whereas sending a WiFi Direct message uses around 1560mW. There are similar energy savings on the receiving end; receiving a Bluetooth message takes 456 mW, while receiving a WiFi Direct message takes 1448 mW [10] this information can be seen in table 2.1. Something that both Bluetooth and Wifi Direct share is that they are both half-duplex technologies [11], [12]. This means that they cannot send and receive messages at the same time. This will play an important role in the throughput of our system when hundreds of messages per second are flowing through the network.

## **CHAPTER 3**

### **RELATED WORK**

Much work has been done in the field of D2D communications and ad hoc networking. Wifi Direct is a newer technology than Bluetooth, so much research is ongoing in this area, while Bluetooth remains one of the most popular communication technologies for short-range communication systems. In this section we summarize some related work in the field of D2D communications, as well as some sources of battery consumption in smartphones.

#### **3.1 Battery Consumption in Smartphones**

In this section we will briefly cover some causes of battery consumption in smartphones. Battery usage is a major concern in smartphone systems, and in [13] and [14] various causes of smartphone battery loss are explored. Each paper introduces different causes of battery loss, including the applications running on the device, the screen, graphics hardware, CPU, user activity etc. What both papers agree on is that the networking interfaces of smartphones play a large role in battery consumption.

#### **3.2 Bluetooth or Wifi Direct Ad Hoc**

There are many papers that explore ad hoc networks using only one technology. In most cases this technology is Bluetooth, but with the rise of Wifi Direct it has become a hot research topic as well. In this section we will discuss two examples of a homogenous ad hoc network of using only Wifi Direct, or only Bluetooth. The two papers discussed here are specific to smartphone ad hoc networks.

Authors in [15] explore the Wifi Direct protocols on Android to investigate the feasibility of creating ad hoc networks with Wifi Direct in real scenarios. They analyze Wifi Direct with two devices and with three devices and provide insight into ways that the D2D connections can be formed.

One of the earliest smartphone ad hoc papers is called Mobiclique [16]. Mobiclique is a social networking application built on top of a smartphone ad hoc network using Bluetooth. This paper was one of the first that was able to show the feasibility of creating mobile ad hoc networks on the fly using Bluetooth.

### **3.3 Heterogeneous Ad Hoc Networks**

In this section we will look into a couple of examples of heterogeneous networks. Heterogeneous networks can be either heterogeneous in device type or communication technology, and an example of each is given here.

In [17] the authors explore the idea of a heterogeneous network of connected devices. They study the internet of things (IOT) and the various networks of connected devices that contribute to the IOT. The authors introduce the typical architecture of these large heterogeneous ad hoc networks and explore ways to improve them. These heterogeneous networks discussed in this paper are built on Bluetooth for communication.

Because there are multiple competing technologies in smart phone ad hoc networks, there sometimes arises issues with devices communicating while running different technologies. For instance, phone A is running Bluetooth, and phone B is running WiFi Direct. If phones A and B want to communicate directly, we run into a problem. An application called BWMesh was developed to attempt to address this problem [4].

The idea behind BWMesh is fairly simple: create a heterogeneous network in which users can communicate even if they are using different technologies. A heterogeneous network is any network that connects devices running different technologies. BWMesh was implemented in the application layer in the form of a simple android chat application that was deployed on multiple android compatible devices. The main idea is that phone A can send a message and it will be sent/forwarded to all devices running BWMesh, regardless of what technology they are using. It relies on at least one node running both Bluetooth and WiFi Direct to act as an intermediary node between devices that cannot usually communicate directly. For example, phone A using a Bluetooth



connection is connected to a phone running both Bluetooth and WiFi Direct, and that phone is connected to phone B running only WiFi Direct. When a message is sent from phone A, it will be sent via Bluetooth to the intermediary phone, who will then forward on via WiFi Direct to phone B [4]. Through use of intermediary nodes, devices that would otherwise be unable to communicate can now send messages to one another. BWMesh was developed as a proof of concept to the fact that different technologies could be combined to create heterogeneous network. There were, however, some issues BWMesh did not address. One of the major issues in all ad hoc networks is energy conservation, and BWMesh did not consider energy needs during development [4].

### **3.4 Switching without Infrastructure**

This section delves into ad hoc networks that employ some sort of switching for energy saving, and does not use any extra infrastructure to facilitate the switching.

In [18] both Bluetooth and Wifi Direct are used in an ad hoc network to conserve energy. Because device discovery is more expensive in Wifi Direct, this paper proposes to do device discovery in Bluetooth, then after connecting to exchange Wifi MAC addresses and create an ad hoc Wifi connection using that information. After the exchange of Wifi MAC addresses all messages are sent by Wifi Direct. They were able to show energy savings over the situation of only using Wifi Direct, however because they are only sending by one technology they are not able to explore full utilization of both technologies.

The authors in [1], do take into account battery concerns when deciding what routing technologies to use. They propose a similar idea to BWMesh in that they create a system that is capable of using different wireless technologies. They implemented a prototype and tested on 30 mobile phones. The motivation for this work was the aftermath of the Great East Japan Earthquake and Tsunami when the authors realized the importance of D2D systems that can be used in the case of network outages. When it came to energy concerns, they decide based on remaining battery level to go into one of two routing "modes," which includes MANET mode and Delay/Disruption-tolerant networks (DTN). This paper, however, does not give an in depth analysis of the energy savings and

limit the mode switching to routing technologies.

### **3.5 Switching with Infrastructure**

This section looks into systems that employ a technology switching scheme, however extra infrastructure is needed to facilitate the switch.

Two examples of communication switching with additional infrastructure are systems called CoolSpots and SwitchR [19] [20]. In both of these papers communication switching between Bluetooth and Wifi are explored with the main goal of energy saving. CoolSpots and SwitchR are both successful in saving energy through switching technologies, however both require additional infrastructure to facilitate the switches, in the form of a communication device that would be kept in your home. While this work is an important step in showing that switching is a viable strategy, they also have a few problems. Mainly, these systems are designed for use in a home with only one device, and are therefore not scalable or able to be used in disaster situations with many devices trying to connect at the same time.

### **3.6 Cross Technology Communication (CTC)**

The papers that follow are examples of systems that utilize two technologies simultaneously through the use of Cross Technology Communication strategies and modification beneath the application layer.

These two papers, BlueBee and WEBee [21], [22] attempt to facilitate communicate between two separate technologies, specifically Bluetooth and ZigBee or Wifi and ZigBee. These papers use Bluetooth and Wifi radios to manipulate the payload of the sending technology to be able be received without error by ZigBee radios. They are able to achieve 99% reliable communication between technologies. While this is reliable, it also requires changes beneath the application layer.

## CHAPTER 4

### DESIGN OF HETEROGENEOUS SMARTPHONE AD HOC NETWORK

In this chapter we will discuss an overview of the HSNet platform. First we will look at an overview of the design of HSNet, followed by an in depth look at the technical details of the project. Finally we will look at the chat application on Android created to test HSNet.

#### 4.1 Design Overview

In this section we discuss our proposed solution to the issue of energy consumption in smartphone ad hoc networks, HSNet. The idea behind HSNet is that mobile ad hoc networks can be enhanced by the use of both Bluetooth and Wifi Direct. Utilizing both technologies improves both the range and device support of the network. The main goal of HSNet is to provide energy savings without sacrificing throughput. To achieve this goal nodes in the network will decide what technology to send messages by based on current energy needs. In addition to this, a load balancing strategy in master nodes is proposed to help create a balanced network topology and save energy on overloaded master nodes. A chat application was built on Android to evaluate our HSNet platform. In this section we discuss some of the design and technical details of HSNet, as well as an example network topology.

While we know that there are benefits and drawbacks to both technologies, we first had to decide what we are trying to prove by exploring the idea further. There is already work done on showing the power and throughput tradeoffs of Bluetooth and Wifi Direct, but we wanted to provide a more fully implemented, real world application to be able to prove our results. Since power consumption is such a major factor, that is where we put our focus. We decided to design HSNet around the idea that we can establish multiple connections, with the ideal technology being Wifi Direct. In times which energy is of higher concern, specifically when the battery life on a smartphone is low, we can sever the Wifi Direct connection and opt for Bluetooth to save energy.

With that goal in mind, there are two main questions we must be able to answer: can we switch

between the two connections on the fly, without data loss, and does this switch actually provide power savings? Most existing work done on comparing Wifi Direct and Bluetooth is done using both separately. This led to our proposed idea of incorporating everything in a single platform. To investigate and gain further insight into how these differences may be used for the benefit of users, we develop a new Android chat application. Android is free to develop for, and runs on a variety of devices. Nearly every new device has both Bluetooth and Wifi Direct capabilities, which makes it an ideal platform to test our HSNet system. The application is written in Java using Android Studio, targeted for devices running API level 23. By creating a full application, we can gather real time data that we believe to be much more valuable than what we could do with a simulator or by simply using estimates of average power consumption.

## **4.2 Technical Details**

In this section we will discuss the three main technical contributions of this thesis. This includes the modified flooding algorithm, the mode switching algorithm, and the mast node load balancing.

The modified flooding algorithm is used for message delivery in the HSNet system, the mode switching algorithm is where the majority of the energy saving is obtained and it facilitates the switching of technologies, and the load balancing system helps prolong the life of the network by creating a more balanced network topology.

### **4.2.1 Modified Flooding**

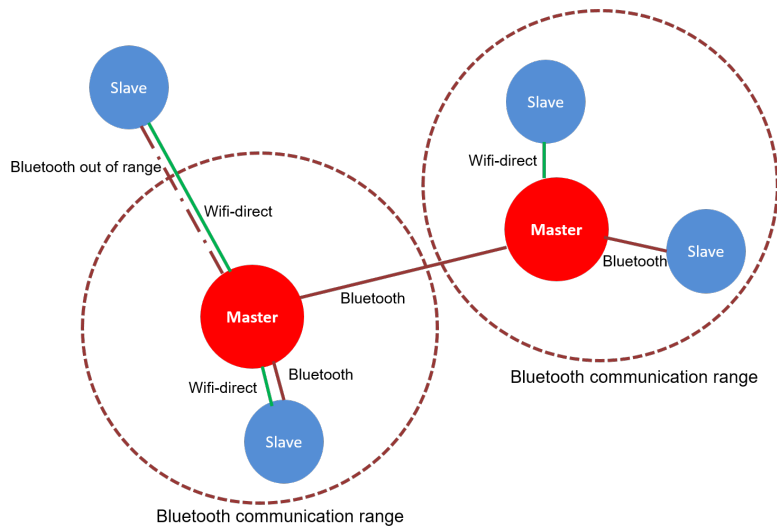
With HSNet, after the network of devices is connected appropriately, a message can be sent from any connected device and it will be received on all other devices in the network. We propose a *modified flooding algorithm* that adds bounding parameters to greatly limit the amount of messages being sent across the network (while still reaching every node). We add a source, direct source (the specific node who sent you the message), sequence number, previously visited nodes, and neighbor fields to our message format, and no message is ever sent if any of these parameters are met. This alone reduces the number of messages being sent across the network by over 1/3 as compared to a

standard flooding algorithm.

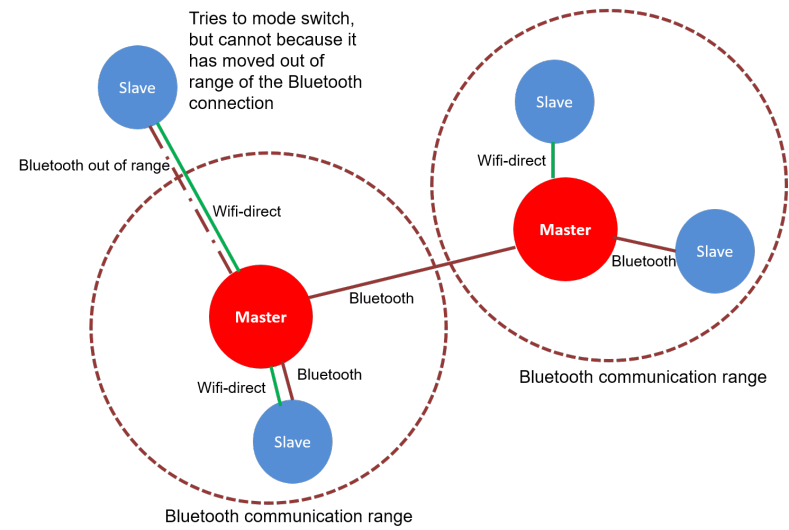
#### 4.2.2 Mode Switching

In most ad hoc networks, the connections between two devices are static in that you will always send via the same technology to the same device. The main energy saving tactic that we employ is a system that will send via Wifi Direct to improve speed and latency, but will choose to send via Bluetooth if a particular node is in need of energy conservation. On each node at any given time it can decide whether to go into a "power saving" mode in which it will send messages by Bluetooth (if available) instead of by Wifi Direct. There are many factors that we had to take into account with our proposed *mode switching algorithm*. On slave nodes in order to do a mode switch you first have to verify that you have a backup Bluetooth connection. In addition to this you have to confirm that you are still in range of this Bluetooth connection. If a slave node meets this criteria then it can sever its Wifi Direct connection in an attempt to extend its life. This solution balances network performance and battery concerns.

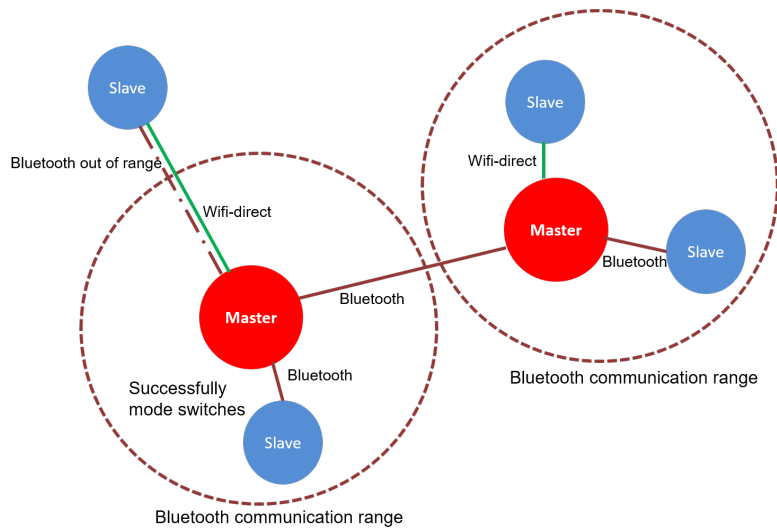
In Figure 4.1 we can see examples of how a mode switch might occur in a real system. Figure 4.1a shows an example network with two piconets connected and two slaves on each master node. Figure 4.1b shows that the slave node in the top left tries to do a mode switch, but cannot because its backup Bluetooth connection is out of range. This node will continue to try to mode switch periodically and if the Bluetooth connection comes back into range then it will be able to successfully mode switch. An example of a successful mode switch can be seen in Figure 4.1c: the bottom slave node on the leftmost master node originally had both a Bluetooth and WiFi Direct connection, but after going into power saving mode the WiFi connection is no longer used. Figure 4.1d shows an example where a slave node tries to do a mode switch, but it does not have a backup Bluetooth connection. In this case it will no longer attempt the mode switch.



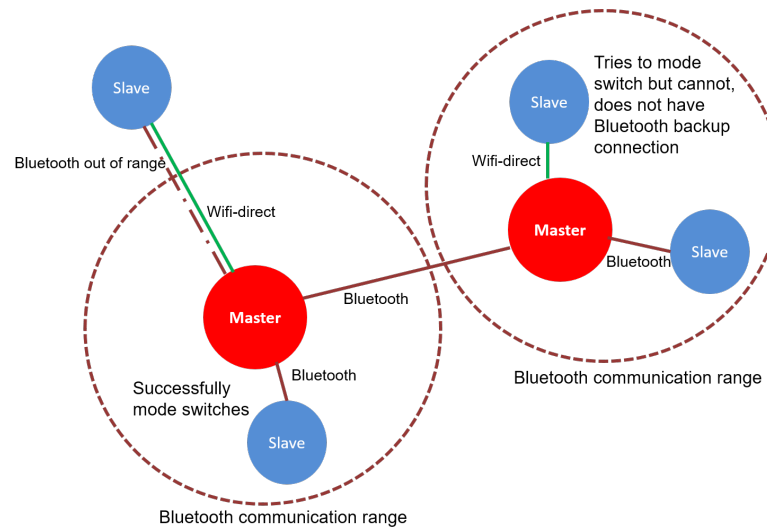
(a) Example network



(b) Bluetooth out of range



(c) Successful switch



(d) No backup connection

Figure 4.1: Example mode switching scenarios

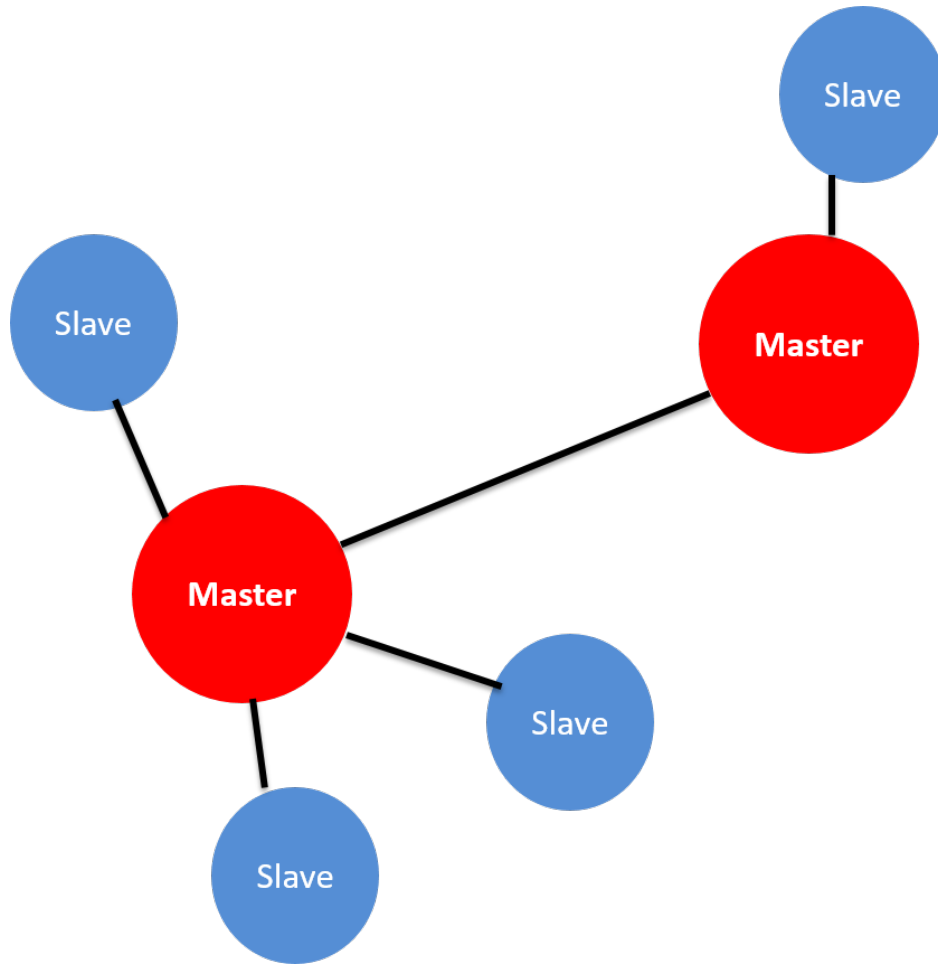


Figure 4.2: Example of an unbalanced network topology

#### 4.2.3 Master Node Load Balancing

On master nodes a similar set of criteria has to be met in order to complete a mode switch. A master node must verify that all of its slaves have backup Bluetooth connections and will remain in range. Master nodes, however, are often juggling many more connections than a slave node, which makes it harder to make a switch to this "power saving" mode. We now run into the problem of the master nodes in the HSNet system consuming much more energy than our slave nodes. This presents a major issue because if a master node dies, then it will disconnect large portions of our network.

Because of this, a *load balancing* system between master nodes is proposed to try and help relieve some of the strain on an overloaded master node. An example of an unbalanced network

topology can be seen in Figure 4.2. In this scenario the master node to the left is overloaded and has three slaves, while the second master has only one slave. For the load balancing system to work, master nodes must routinely send status messages to other master nodes in the system. These status messages are sent as normal messages and discarded at any node that is not a master node. These status messages include information about how many slaves are currently connected to the master. If an imbalance is detected, then the overloaded master node will signal to its slave nodes that, if possible, they should switch master nodes. Only one slave at a time can switch, and will only switch once it has confirmation from its current master. The caveat with this feature is that the slave node must have been previously connected to the other master node, otherwise this system cannot work, which is a limitation of the Android APIs regarding Bluetooth and Wifi Direct. This load balancing algorithm present within HSNet greatly improves the battery life of master nodes, and alleviates some major concerns regarding network life.

Let's take a look at an example of how the master node load balancing might happen. As we can see in Figure 5.4a the master nodes routinely send status messages to each other to exchange information on number of slaves. In Figure 5.4b you can see that an imbalance was detected and the overloaded node sends a switch message to all of its slaves. The switch is agreed by one of the slaves in Figure 5.4c, and in Figure 5.4d after the switch has occurred the network is now balanced.



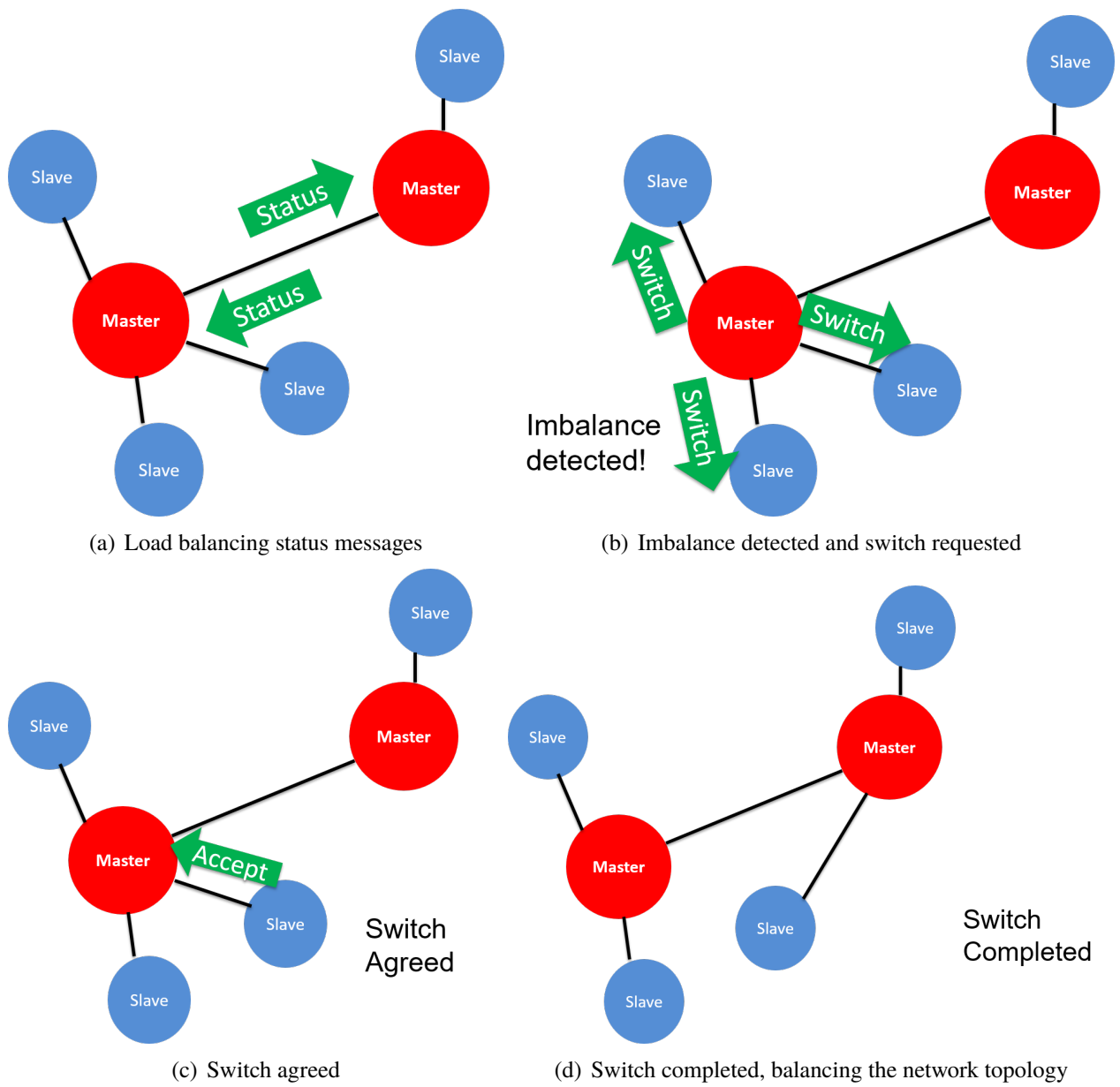


Figure 4.3: Load balancing of master nodes

### 4.3 HSNet Chat Application

In this section we explore the chat application running the HSNet system and explore various features and uses. When first entering the app, the user is greeted by an empty chat screen. There is a menu that will bring up various features and options of the HSNet system, this is shown in Figure 4.4.

The connect Bluetooth menu option presents a list of available devices to connect with via Bluetooth, and similarly, the menu option for Wifi Direct presents a list of devices that are available to connect with via Wifi Direct. Additionally, there is a menu option to disconnect Wifi manually. This would be used if the user wants to manually switch to its Bluetooth connection instead of waiting for a certain battery level to automatically shut off Wifi via the mode switching algorithm present in HSNet.

There is also an option to switch the battery threshold for the automatic switch from Wifi Direct to Bluetooth. The app is coded to switch from Wifi Direct to Bluetooth when a node reaches 30% battery, but if for any reason the user wants to change that value, the menu option to change the threshold will all the user to select a new switching threshold.

For the purposes of testing the limits of our system as well as introducing additional battery strain, an option menu to automatically send 20 messages per second is given. In addition to the autosend there is also a menu option to stop a currently running autosend. Another option menu is given to update the autosend rate. As will be seen in the Performance Evaluation section of this thesis, many different sending rates were tested. The autosend feature is what was used while collecting data to test whether our system performed better for battery consumption than traditional ad hoc solutions.

The screenshot shown in Figure 4.5 demonstrates multiple features of our system. First, you will notice that the auto send feature is being utilized. The automatic switching from Wifi Direct to Bluetooth has occurred and is shown. You can tell whether something was received by Wifi Direct due to the presence of the value true immediately preceding the sequence number of the message. If a message was received via Bluetooth the value will be false. The screenshot shows that after

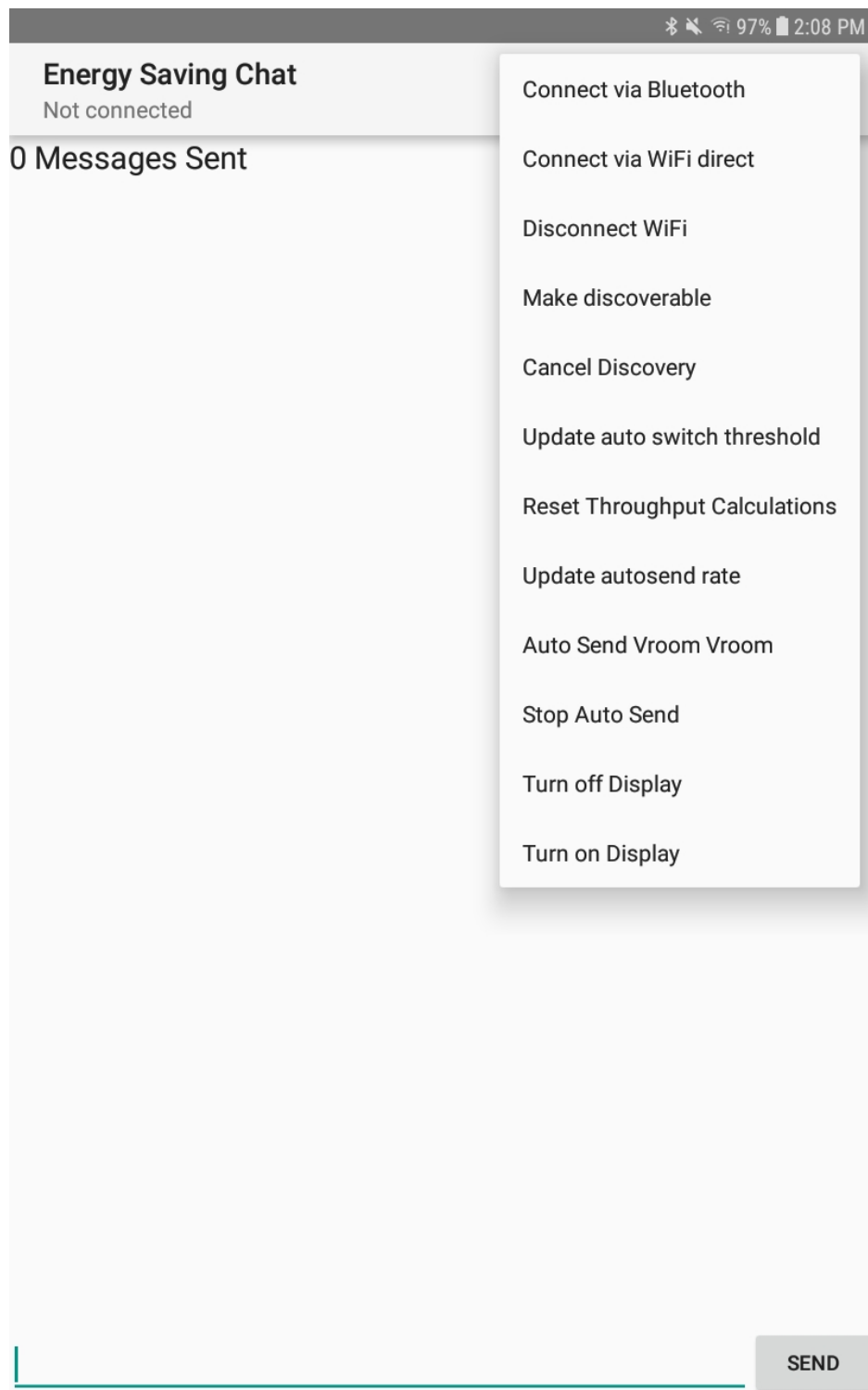


Figure 4.4: Menu showing the different options and settings of our system

message 71 was received, the Wifi Direct connection was severed, and every message afterwards has the value "false," which indicates that it was received via a bluetooth connection.

## 4.4 Network Topology

The network graph in Figure 4.6 shows a standard network structure that was used for testing. Unless noted, the rest of the work will assume a network topology as seen in Figure 4.6. In this example, there are two sub piconets shown. Node D is a Wifi Direct group owner, connected via Wifi Direct to nodes A, B, and C. Wifi Direct connections are shown as a red arrow, whereas bluetooth connections are shown by a dashed blue arrow. Node E is another Wifi Direct group owner and is also bluetooth master. Nodes F and G are connected to Node E via Wifi Direct, but node H is in power saving mode and has severed the Wifi connection in favor of Bluetooth.

These two piconets can be connected in two different ways. They can be connected via a connection with the master node of another piconet, or via a connection with a slave of a different piconet. The connections between master nodes will try and use bluetooth if possible because the battery strain on master nodes is significantly greater than the battery strain of slave nodes. If a Bluetooth connection is not possible, then Wifi Direct will be used, but this should be avoided. For this example, we can see that node D is connected to node E via a Bluetooth connection. It could, of course be possible for node A to provide the link between the piconets, but for this example the piconets are connected via master nodes.

Now let's look at how a message will be passed through the network. When node A sends a message, it sends it along its WiFi Direct connection to group owner D. Node D then forwards the message to all of its children, except for node A, the source of the message. Node D is actually a slave of node E, so it will also forward to node E. After D sends the message along its Bluetooth connection to E, node E will then send via Wifi Direct connections to nodes F and G, and will send via a Bluetooth connection to node H. This general flow through the network holds for sending from any source to all destinations.

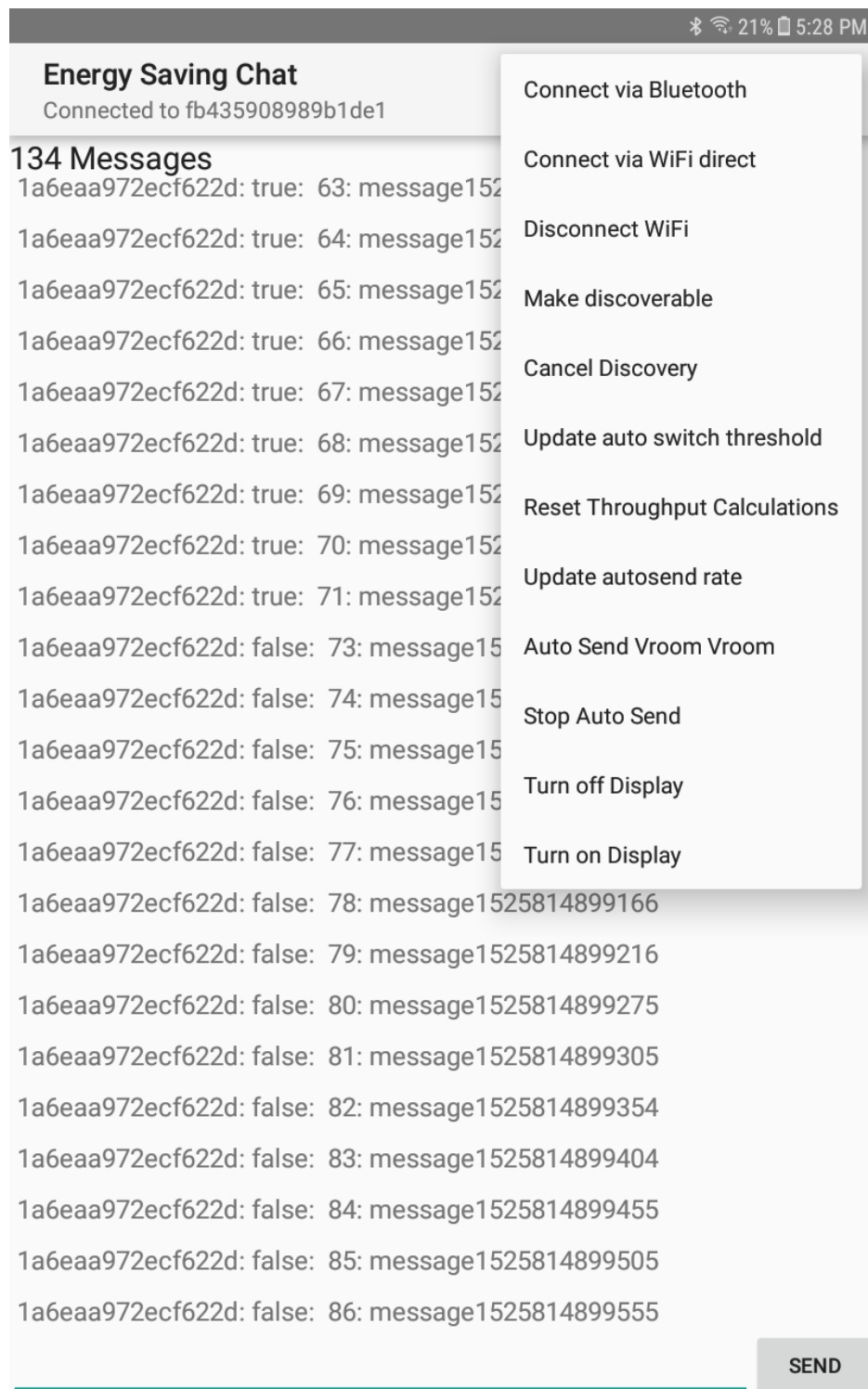


Figure 4.5: Screenshot of our application that showcases our autosending feature, the switching of technologies, and the options menu

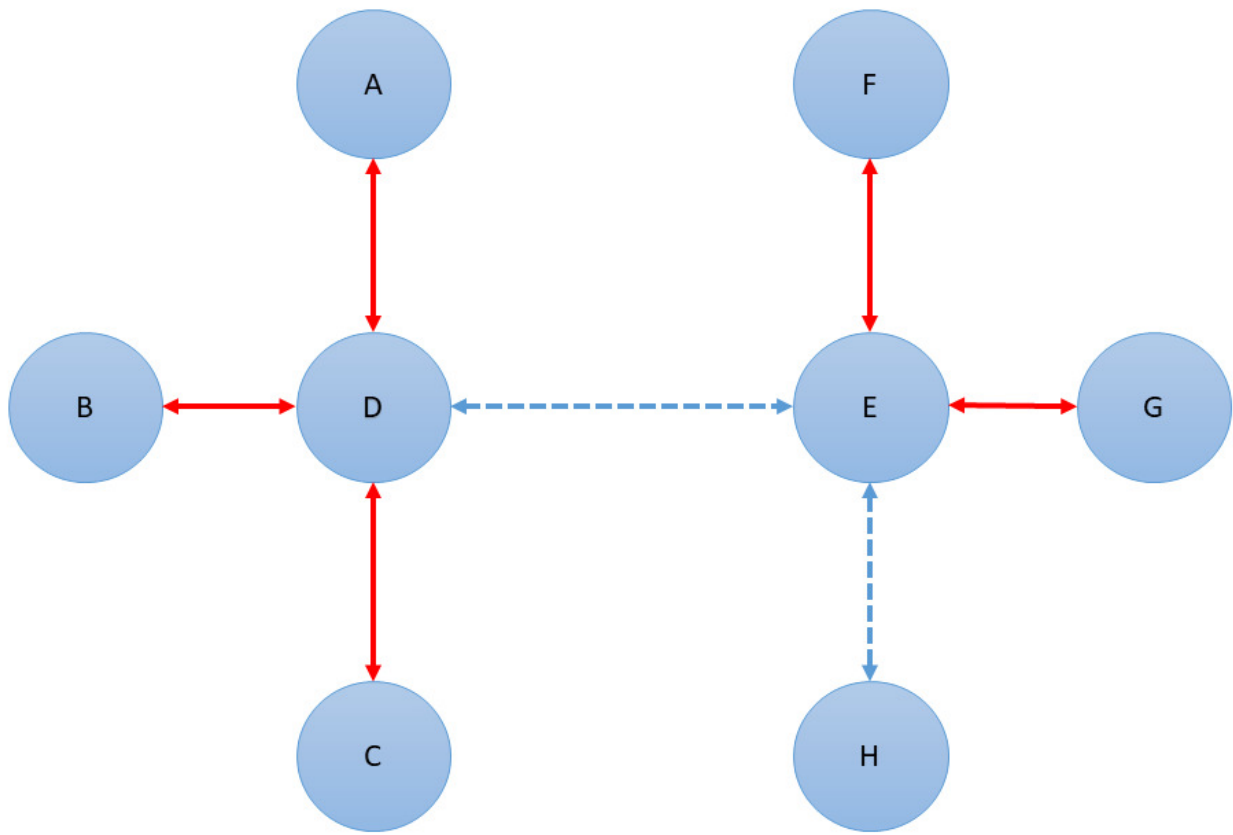


Figure 4.6: Example network topology

## CHAPTER 5

### PERFORMANCE EVALUATION

This chapter is focused on a quantitative evaluation of the HSNet system. We will focus on energy savings for slave nodes, master nodes, as well as look at the overall throughput of the system.

#### 5.1 Evaluation Methodology

In this section we will explore the experimental testbed used to run and collect data about the performance of the HSNet system. We also discuss the different metrics used to evaluate the HSNet system that will be shown throughout the rest of this chapter.

##### 5.1.1 Testbed

Once the application was designed, implemented, and tested for functionality, we set to work gathering data. In this section we outline and discuss tests we have run to evaluate HSNet in regards to energy consumption of both slave and master nodes, as well as the throughput of our system in different configurations and situations. All tests were run on brand new, fresh out of the box Samsung Galaxy Tab A 8.0 inch tablets. All possible sources of energy consumption were standardized across the devices, such as screen brightness, background apps etc. The reason for using the same devices is to provide consistency. Using different types of devices might skew results because of varying factors such as resting battery consumption as well as total battery life. Outside of the tests outlined in this work, the application has been tested on a multitude of different android devices and can support any devices that run either Bluetooth or Wifi Direct.

Since our main focus is power saving, we needed some way to accurately measure the energy usage. To do this, we opted to use the free application *Trepp Power Profiler*, created by Qualcomm [23]. The app runs over top of any application and collects data on a variety of components, including cpu usage, network state, and most importantly, battery consumption.

The three main areas of the evaluation of the HSNet platform are the analysis of energy

consumption of Slave nodes, the energy consumption of Master nodes, and throughput analysis of the system. To run these tests our HSNet software was installed on all of the tablets and were connected with different network topologies based on the different scenarios. The analysis in this work is based on a network topology that was shown previously in Figure 4.6.

### 5.1.2 Evaluation Metrics

HSNet has many different facets that all need to be explored. We evaluate our system using the following metrics.

- Slave Node Analysis
  - Energy consumption of slave nodes is measured in  $\mu\text{W}$  and is the total energy consumption of the device. We show three scenarios of a connection by only Bluetooth, a connection by only Wifi Direct, and a connection that switches half way through the test as an example HSNet connection.
  - Lifetime of the network is measured in minutes and is used to compare how long a Bluetooth network, Wifi Direct network, and an HSNet network last under the same conditions.
- Master Node Analysis
  - Energy consumption of master nodes is measured in  $\mu\text{W}$  and we look at how much energy the typical master node uses compared to a slave node, and we also analyze an example of the master node switching.
- Throughput
  - Throughput of our system is defined as the number of messages received in a given time frame. We compare this number to a theoretically perfect system in which every message is received with no loss in order to evaluate the viability of HSNet. We test



throughput at multiple different message sending rates from 100 messages per second up to 500 messages per second.

- We look at a few different situations of throughput including messages sent within one piconet, and also messages sent between piconets. In the case of multiple piconet communications the message has to travel through more nodes to get to its destination. Again, these tests are run at different sending rates from 100 messages per second to 500 messages per second increasing at intervals of 50 messages per second.
- Lastly we look at how throughput of our system is affected by the battery percentage at which we perform the mode switch. We compare the total throughput of the system when the mode switch occurs at 0%, 20%, 40%, 60%, 80%, and 100%. We compare these values against one another as well as measure them as a percentage of the throughput of a perfect system. For example, if in a perfect system 100 messages are sent and received and our system sends and received 65 messages, then the value for that combination of sending rate and switching threshold is 65%.

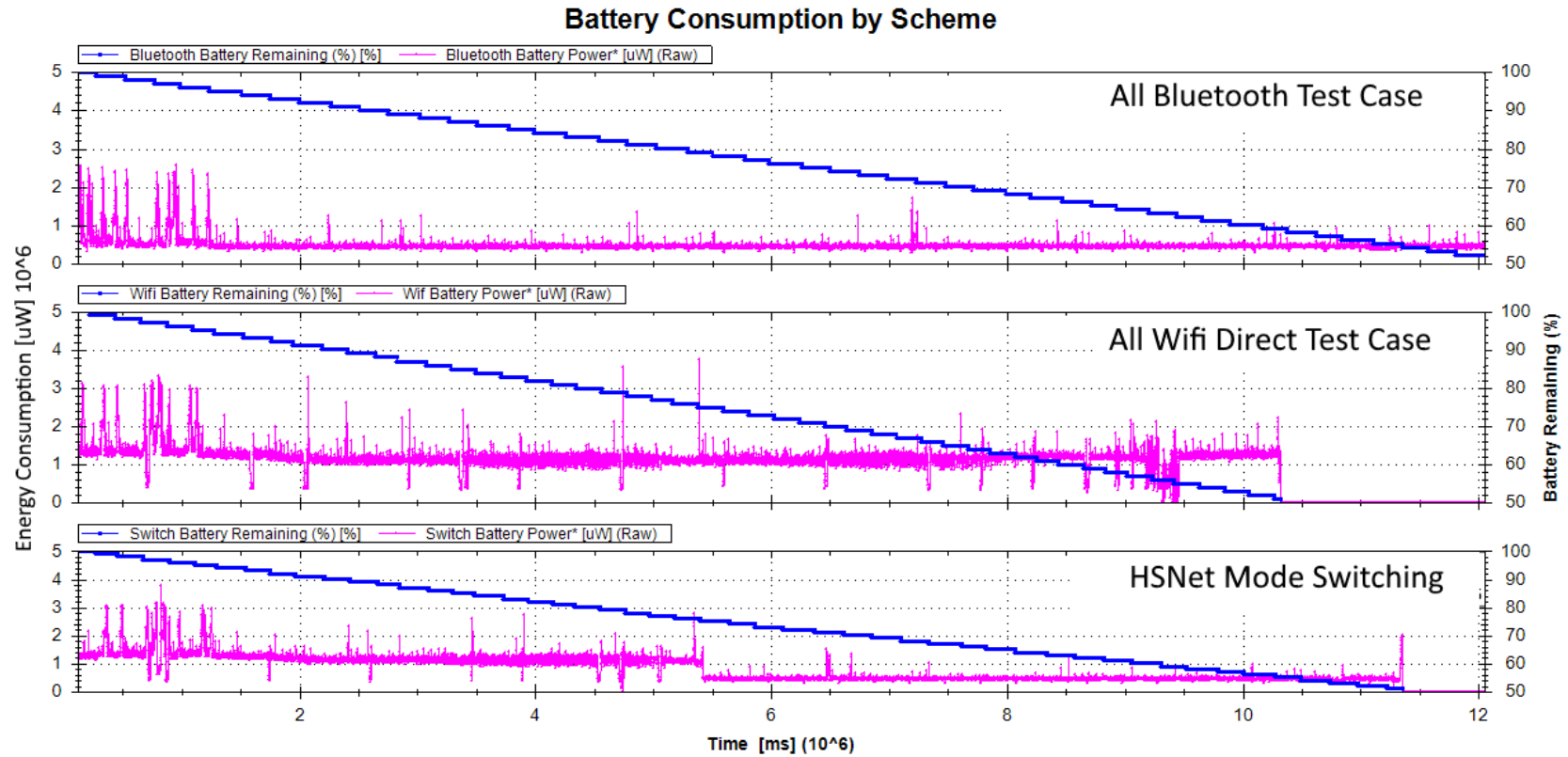


Figure 5.1: The results of our experiments regarding energy consumption of slave nodes. The top graph is a slave node sending by only Bluetooth. The middle graph shows a slave node sending by only Wifi Direct. The Bottom graph shows a slave node that uses our technology switching paradigm for battery savings

## 5.2 Slave Node Analysis

In this section we look into energy consumption as it relates to slave nodes in our system. We gathered information on how much power our application requires in three different scenarios. The first scenario is a network connected entirely by Bluetooth, the second test is a network connected entirely by Wifi Direct, and the third test utilizes our proposed HSNet system and shows our mode switching algorithm being used. All three of these tests can be seen in Figure 5.1. The goal of these tests was to calculate average power consumption, as well as how long the battery will last in the different situations. For these tests we ran each device until it went from 100% battery down to 50% battery. Each test was run simultaneously as three slave devices to one master node.

First we ran a control test in which we used the autosend to send messages exclusively over Bluetooth. The device used an average of  $466,155.3 \mu\text{W}$  and lasted 3 hours and 23 minutes, these values can be seen in Table 5.1. This can be seen in the top graph of Figure 5.1. Next we ran another control test in which the device only sent messages via Wifi Direct. In this scenario the device used an average of  $1,002,054 \mu\text{W}$  and lasted 2 hours and 52 minutes. This can be seen in the middle graph of Figure 5.1. It is notable that the device running only Wifi Direct was using, on average, over twice as much energy as the device running only Bluetooth.

Then we ran a similar profile, except this time the application was using HSNet and switched from sending via Wifi Direct to sending via Bluetooth. For this test we set the switching threshold to 75% battery remaining, which is equivalent to half of the total battery life, as we were running the devices from 100% battery to 50% battery. This is shown in the bottom graph of Figure 5.1. As can be seen in the figure, there is a very distinct transition from where the Wifi Direct disconnects and the messages begin sending over the Bluetooth connection. During this test the device used an average of  $761,068.2 \mu\text{W}$  and lasted 3 hours and 9 minutes. The average battery consumption was almost exactly halfway between the baseline tests, and shows that switching from Wifi Direct to Bluetooth is a viable option for energy saving. Also shown in the figure is the total lifetime of the device within the scope of the tests. You can see when each device ran out of power as the average battery consumption drops to zero. The third test where our HSNet switching mechanism

Table 5.1: Energy consumption and lifetime analysis of slave nodes

	Average Power Use	Total Lifetime
All Bluetooth	466, 155 $\mu$ W	203 minutes
All WiFi Direct	1,002,054 $\mu$ W	172 minutes
HSNet	761,068 $\mu$ W	189 minutes

was introduced shows significantly longer battery life than the Wifi Direct case.

There are a few very key points that need to be recognized with this third test. As shown in Figure 5.1 when this third test was sending via Wifi Direct it was consuming almost exactly the same amount of power as the test where messages were only being sent by Wifi Direct. After the switch to Bluetooth, the device was using almost the exact same amount of power as the test where only Bluetooth was used. This is significant because in the third test, both the Bluetooth and Wifi Direct connections were active while sending by Wifi Direct. This shows that having these extra Bluetooth connections as a backup do not actually consume relevant amounts of energy if no messages are being sent. One of our concerns was the potential of doubling the actual amount connections between devices, but this is a non-issue as inactive connections consume trivial amounts of energy.

An interesting observation is that you can see an almost identical spike in power consumption at the beginning of each test case. This is due to the power overhead needed for device discovery and to set up the initial connections. The third test where both Wifi Direct and Bluetooth were connected does not show a larger or more prolonged spike during this connection period, which is something that will be discussed later when analyzing power consumption on the master nodes.

It is important to note that these power consumption numbers are for the entire smartphone, not only our application. That is why we took steps to ensure that outside forces would have minimal impact on energy usage.

### 5.3 Master Node Analysis

In this section we discuss the energy consumption when it comes to the master nodes in our system. The master nodes in our system differ greatly compared to the slave nodes. A slave node can only send to its master node, and therefore will consume a somewhat fixed amount of energy.

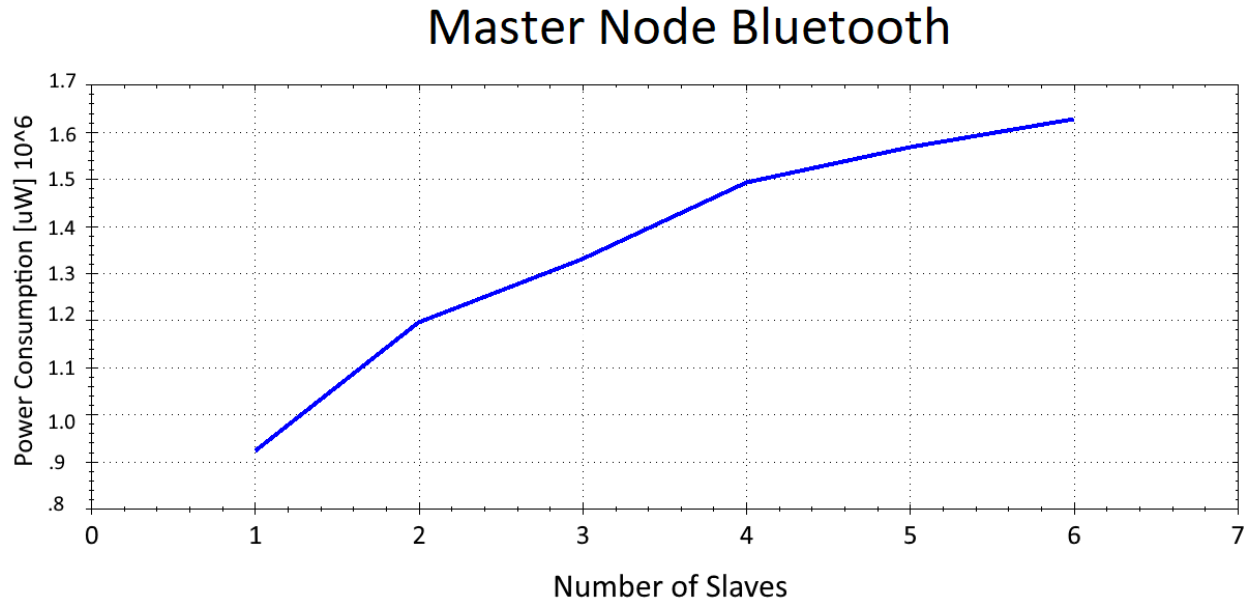


Figure 5.2: Power consumption of a Bluetooth master node with respect to the number of slaves

The master nodes, however, can have up to seven slaves in the Bluetooth system and four slaves in the Wifi Direct system. The master node has to send to all of these slaves, which consumes significantly more energy than a slave having to only send to its master. In order to combat this issue, we look at how much energy the master node is actually consuming with the addition of more slaves, and we also propose and implement a *load balancing* system in which slave nodes will switch from one master to another if there is an imbalanced network topology.

First we look at how the energy consumption of the master node increases with the addition of more slaves. We ran tests and took the averages of having one slave all the way up six slaves on a Bluetooth master, and having one slave all the way up to four slaves on a Wifi Direct group owner.

Figure 5.2 shows how the energy consumption increases as the number of slave devices increases on a Bluetooth master node. We see that while energy consumption does increase with the addition of slave nodes, the increase, however, is not linear. We start to see the energy consumption leveling off a bit with more slave nodes being added. This is important to our system because the potential strain on the master node is increasing at a less severe rate when slave nodes are added. This happens, and was discussed briefly in the previous section, because the power overhead needed to

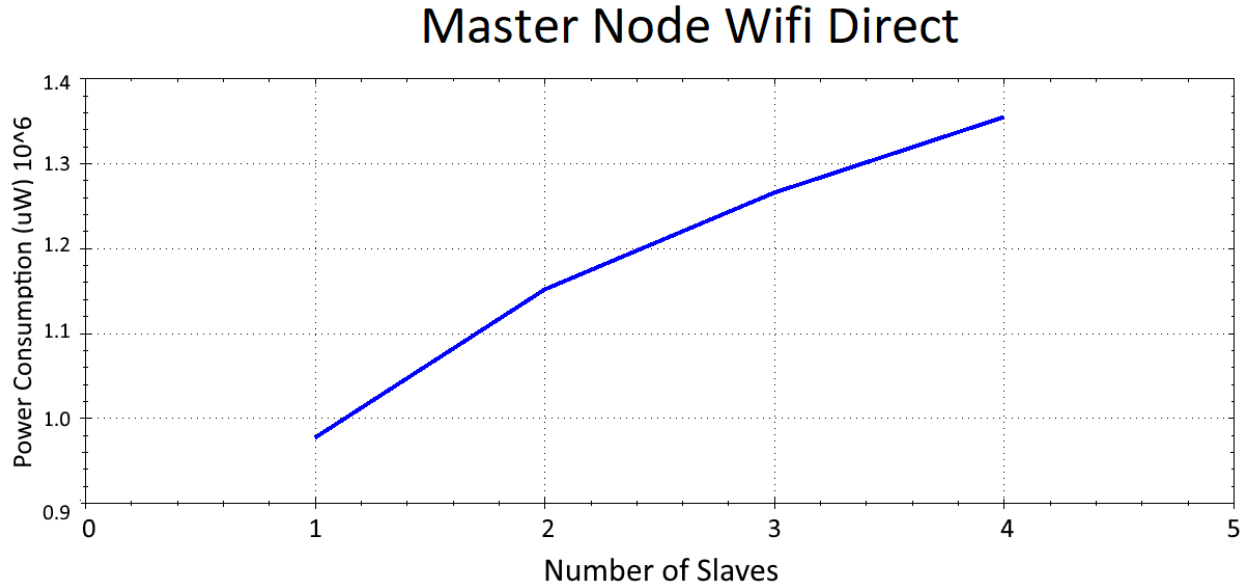


Figure 5.3: Power consumption of a Wifi Direct group owner with respect to the number of group members

initially set up the connections does not increase significantly with more than one connection. We see in the bottom graph of Figure 5.1 that two connections did not produce a noticeably larger spike in energy consumption than one connection. We ran the exact same tests with Wifi Direct group members being added to the group owner, and got similar results, showing that adding slaves to a master node in either system behaves the same way. A similar graph is shown with Wifi Direct group members being added to the group owner in Figure 5.3.

Even though the addition of more slave nodes does not produce a linearly increasing amount of energy consumption, the master node does consume more energy than a standard slave node. As we can see in Figure 5.2 with six slaves the master consumes around 1,600,000  $\mu\text{W}$ , which is a little bit less than four times as much energy as a slave running Bluetooth will consume on average.

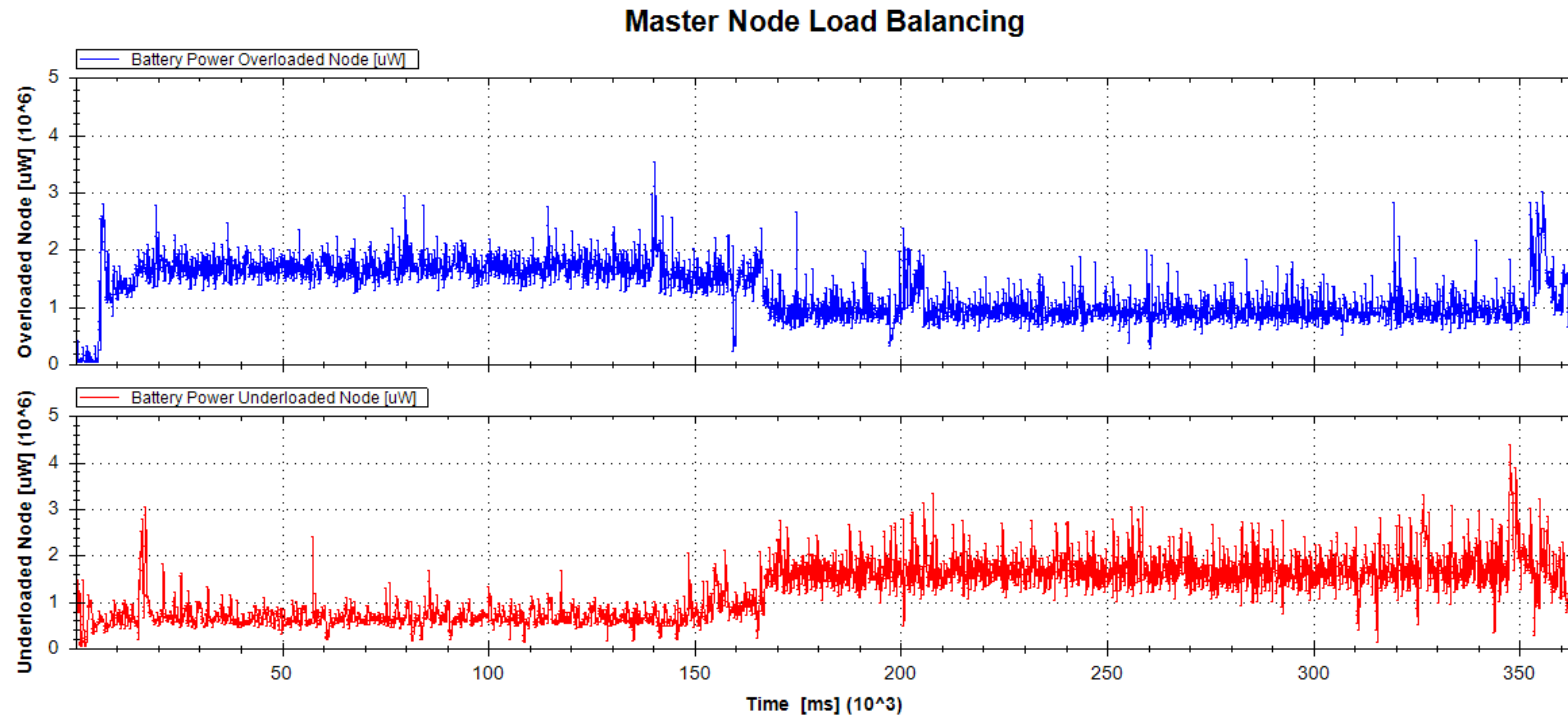


Figure 5.4: Results of an experiment to showcase the load balancing system for master nodes

Table 5.2: Energy consumption of master nodes before and after load balancing

	Before Load Balancing	After Load Balancing	Average Consumption
Overloaded Master	1,613,624 $\mu$ W	990,317 $\mu$ W	1,274,904 $\mu$ W
Underloaded Master	713,419 $\mu$ W	1,707,874 $\mu$ W	1,251,682 $\mu$ W

Figure 5.4 shows the results of a test showcasing the load balancing system of HSNet. The top graph of the figure shows a node that originally had two slaves, and the bottom graph shows a node that originally had one slave. As is shown, the master nodes realize that there is an imbalance in the network topology as the "overloaded" node has twice as many slaves as the "underloaded" node. After this has been realized by the master nodes and a switch has been agreed, you can see a simultaneous drop in energy consumption on the overloaded node, and a jump in energy consumption on the underloaded node. Before the load balancing occurred, the overloaded node was using 1,613,624  $\mu$ W, while the underloaded node was using 712419.5  $\mu$ W, these values can be seen in Table 5.2. After the switch occurs, the previously overloaded node is now running at 990,317.4  $\mu$ W, and the previously underloaded node is running at 1,707874  $\mu$ W. This is quite a difference in terms of energy consumption on these two master nodes before and after the load balancing. Overall, the two nodes had an average power consumption of 1,275,904  $\mu$ W and 1,251682  $\mu$ W, which is relatively balanced, considering the load balancing occurred roughly half way through the test.

If this system were allowed to keep running, the load balancing would continue, because the master nodes are again unbalanced after the switch. That same node that did the initial switch will switch back to the original master after the next round of master status messages are sent. This type of load balancing where the same node keeps switching between masters will occur in all situations where one master node has only one slave more than the second master node until both masters have at least four slaves. In situations where the overloaded master has only one more node than the underloaded master but both have four or more slaves the energy overhead needed to do the switching outweighs the energy savings received on the underloaded node. If, however, the situation was three slaves on one master and one slave on the other. After the first switch



there would be no more switching as the master nodes are now balanced. As a note, these energy consumption numbers may seem higher than a standard master node because in this situation we are sending 300 messages per second, which is a higher rate than in other tests.

## **5.4 Throughput Analysis**

In this section we discuss the throughput of our HSNet system. In the case of our application, throughput is in reference to how many messages we can successfully receive in a window compared to some theoretically perfect system.

Throughout the first sections of this thesis we have been discussing ways in which to conserve battery life of our network. If all we care about is battery consumption, then Bluetooth would be the obvious choice. Wifi Direct, however, is a popular choice for ad hoc networks because of its extended range, and its significantly better throughput compared to Bluetooth. There are situations where we will need to decide what is the best switching threshold if high throughput is something needed by the system. To test this, we look into how our system compares with the baselines of only sending by Bluetooth, only sending by Wifi Direct, and how it compares to a perfect system in which every message can be received with no latency issues.

We will again use the network topology shown in Figure 4.6. In this group of tests, node A is sending messages at different rates, shown in the x-axis of Figure 5.5, and we measure how many can be received at node C in 60 seconds. The messages will travel from the slave node A, to the master node D, and arrive at the destination node C.

We ran this test three times for each different message sending rate. We ran the two baselines of sending exclusively by Bluetooth or Wifi Direct, as well as having one of the two connections be Bluetooth and the other connection being Wifi Direct. This third test is an example of a heterogeneous network using HSNet.

## Average Throughput A->C

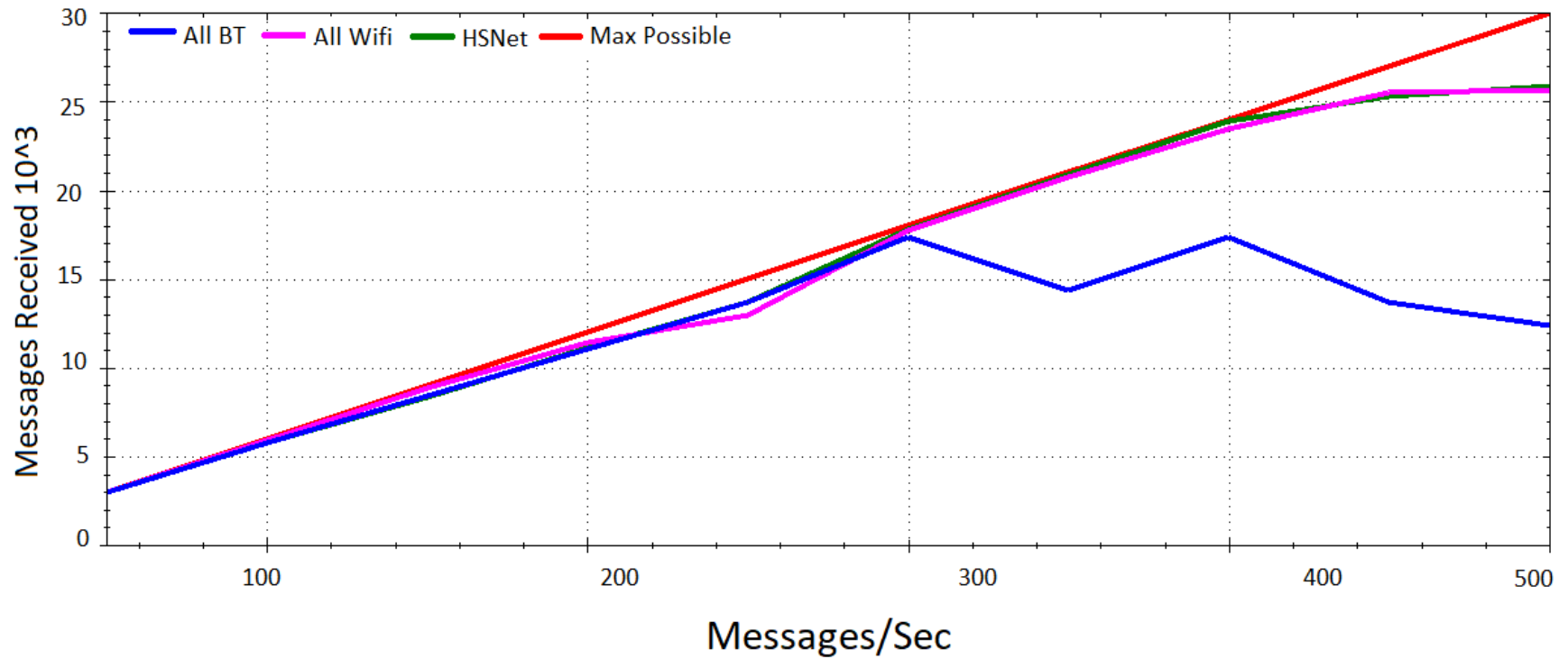


Figure 5.5: Throughput analysis of communication within a single piconet

Figure 5.5 shows the results of this first test. The red line is the maximum possible number of messages received given a perfect network. The interesting thing to note in this figure is that the pink and green lines indicating all Wifi Direct and HSNet connections respectively show almost identical throughput numbers. Both of those situations hold close to the max possible until about 400 messages per second, at which point they level off. On the other hand, the all Bluetooth connections can match the Wifi Direct and HSNet connections until about 300 messages per second at which point it levels off. This means that at a rate of less than 300 messages per second, a network connected entirely by Bluetooth shows the same throughput as a network connected only by Wifi Direct and also the same throughput as the heterogeneous network.

In the second group of tests shown in Figure 5.6 a similar situation was explored. In this test we are measuring how many messages can be sent from node A and received at node F in 60 seconds. The main difference in this scenario is that node F is in a different piconet than node A and the message will have to travel along more nodes to reach the destination. We again ran this test for different message sending rates, but this time we ran the test four times, for each scenario of: all Wifi Direct connections, all Bluetooth connections, the first connection of A→D being Wifi Direct and E→F being bluetooth, and finally the first connection of A→D being bluetooth and E→F being Wifi Direct. These last two scenarios are two different examples of connections possible in the heterogeneous network using HSNet. It is important to note that in this scenario the connection between nodes D and E is always bluetooth as the connections between master nodes do not usually switch.

The results in Figure 5.6 show that, once again, the situation of all Wifi Direct connections performs as well as the maximum possible until around 400 messages per second, at which time it levels off. We can also see that a mix of Bluetooth and Wifi Direct connections can boast near perfect throughput until about 250-275 messages per second, at which point both situations of HSNet connections level off. However, the situation of all connections as Bluetooth hits its max throughput after about 100 messages per second.

## Average Throughput A-F

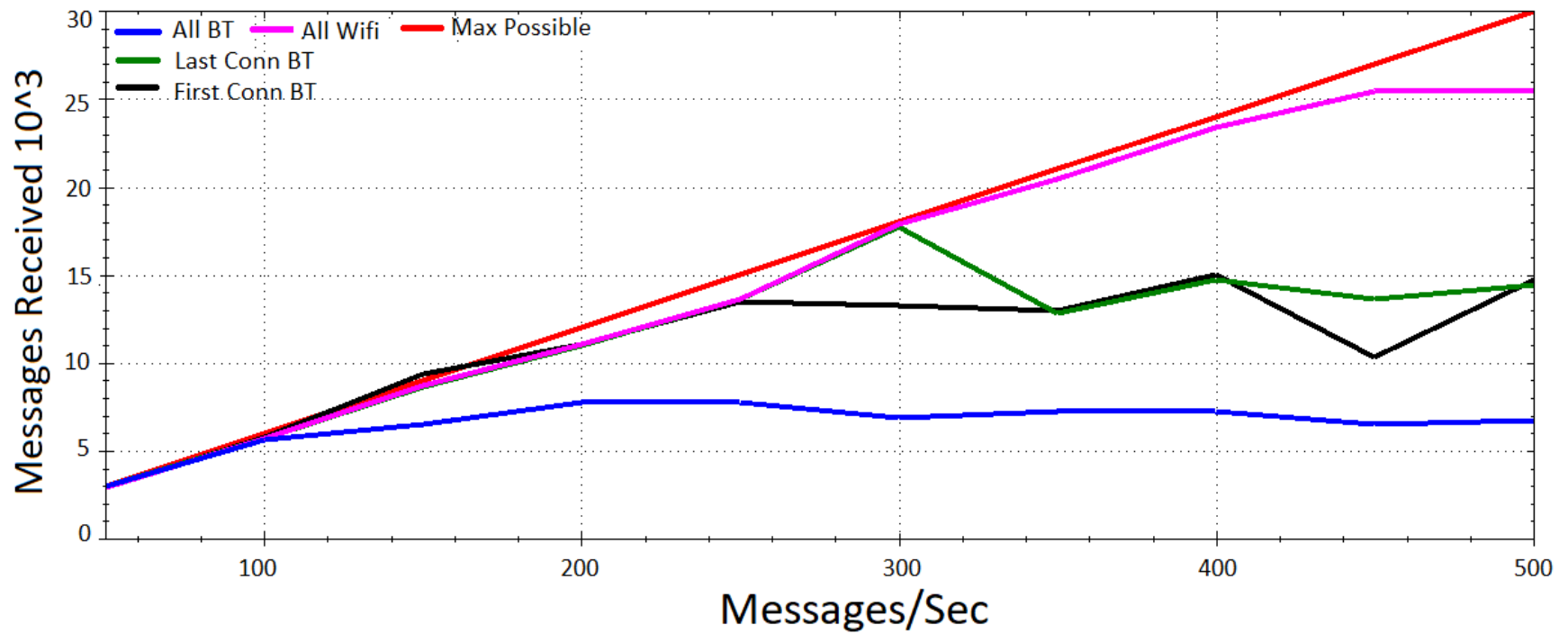


Figure 5.6: Throughput analysis of communication between two piconets

The difference in throughput between Wifi Direct connections and Bluetooth connections is much more evident in the situation of sending messages over more nodes and between two piconets. Anything less than 300 messages per second will show minimal differences if half of the connections are Wifi Direct and Bluetooth, in which case any switching threshold would yield similar throughput results. However, if switching to Bluetooth results in a network where all of the connections are Bluetooth, then it can only maintain throughput close to maximum at 100 messages per second and lower.

The last group of tests is shown in Figure 5.7. In this scenario we are sending messages from node A to node F. This test assumes that all connections are originally Wifi Direct, and the connection between D and F is going to switch from Wifi Direct to Bluetooth at some threshold. In this test we analyze the effect on throughput at different switching thresholds.

As can be seen in the bar graph of Figure 5.7 we have the switching threshold on the x-axis. This is 0% all the way up to 100% by intervals of 20%. A switching threshold of 0% means that once the phone reached 0% it would switch to Bluetooth. In that case it would never switch, and in the case of switching at 100% you would switch immediately. For the ones in between, however, such as a switching threshold of 40% means that when there is 40% battery left the switch from Wifi Direct to Bluetooth occurs.

We measure the throughput as a percentage of the maximum possible for 100 messages per second up to 500 messages per second, at increasing intervals of 100 messages per second. The closer to 100% means the average throughput for the different sending rate and switching combination was closer to the maximum possible.

As can be seen in the figure, for sending rates of 100 and 200 messages per second no matter what the switching threshold is, the average throughput of the system remains close to the maximum possible. This indicates that if less than 200 messages per second are flowing through the network then the switch from Wifi Direct to Bluetooth can occur with no degradation of throughput. At 300 messages per second we start to see a significant drop in overall throughput once you reach a switching threshold of 80%. This means, however, that a switching threshold of 60% or lower

will not result in significantly less throughput at 300 messages per second. At a sending rate of 400 messages per second we see almost the exact same trend as with 300 messages per second. Which says that at 400 messages per second you can have a switching threshold of 60% or lower with minimal throughput losses. The case of 500 messages per second, however, shows significant drops in throughput once you hit a switching threshold of 20%. This indicates that if messages are flowing through the network at 500 messages per second, then switching from Wifi Direct to Bluetooth will always result in a loss of overall throughput.

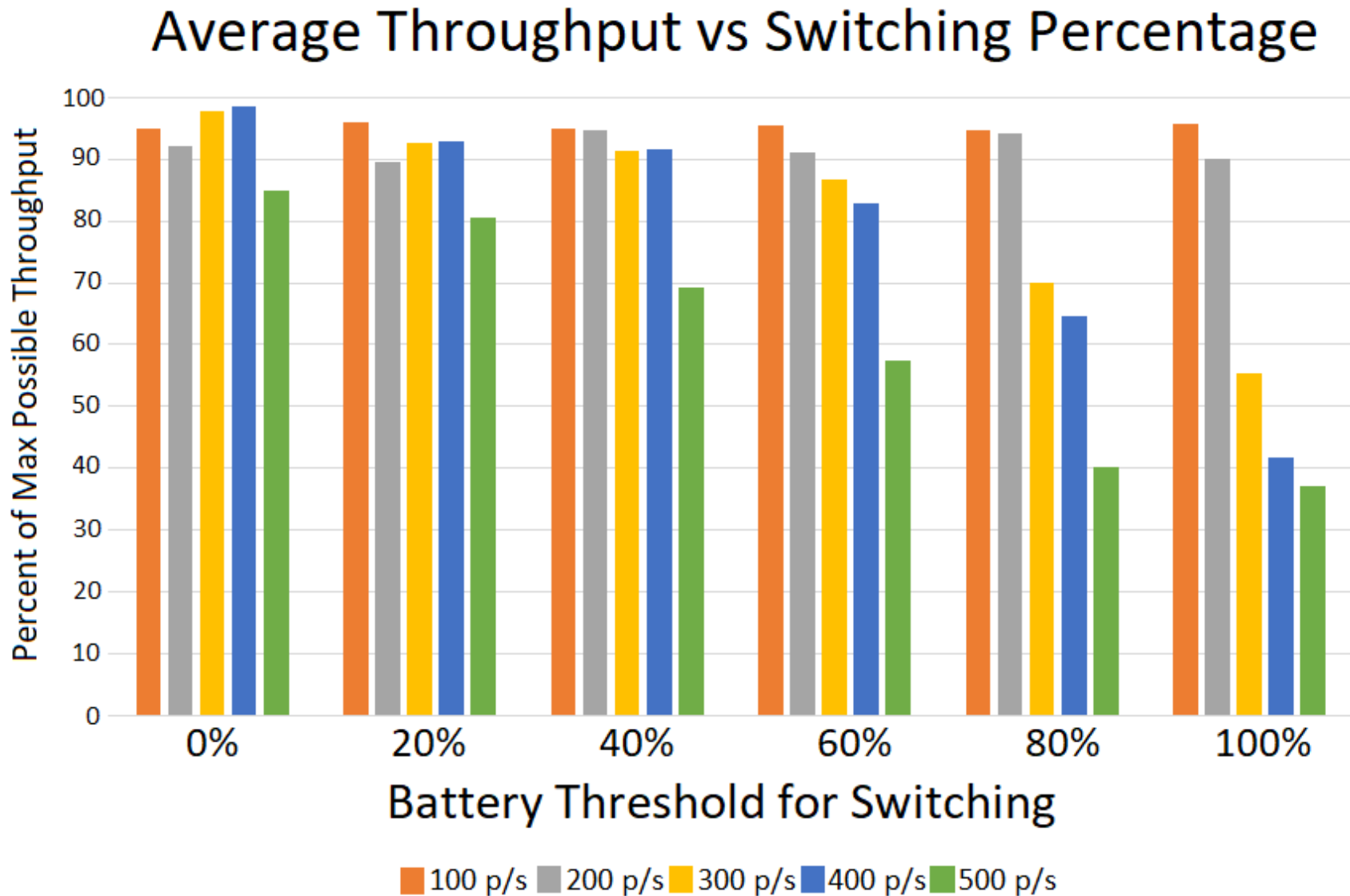


Figure 5.7: Throughput analysis with respect to the threshold for switching technologies

## CHAPTER 6

### CONCLUSIONS AND FUTURE WORK

This chapter will conclude this thesis as well as look at potential future work to advance the HSNet platform.

#### 6.1 Conclusion

In this work we introduce a multi-hop heterogeneous smartphone ad hoc network, HSNet, to utilize both Bluetooth and Wifi Direct that can send messages over each of the technologies individually, as well as automatically switch from Wifi Direct to Bluetooth on the fly.

We analyze the power consumption of networks that communicate exclusively via Wifi Direct or Bluetooth, and compare it to HSNet that communicates seamlessly over both Wifi Direct and Bluetooth. We show that our system of switching technologies in real time can significantly increase the lifetime of the network. We look individually at the power consumption issues with slave nodes, in which the solution of switching technologies shows significant power reduction.

Master nodes are under significantly more battery strain because they have so many more connections. We show that having more connections consumes more energy, but this does not mean a linear growth in power consumption. We introduce a load balancing strategy for master nodes to help alleviate the issues with having so many connections.

Lastly, we discuss in depth the tradeoffs between throughput of our network and battery consumption. In many cases where messages are sending at less than 300 messages per second we see almost no degradation in throughput with respects to what technology is being run on a given node.

Mobile ad hoc networks are becoming increasingly common as the number of mobile devices grows rapidly. Since saving energy is crucial when working with mobile devices, our system is a step towards efficiently managing a network by taking into concern that the battery of some devices may be low and energy is more critical.



## **6.2 Future Work**

Future work for this project will be focused on two main areas: routing and simulation/scale. Currently in our system message delivery is done through a modified version of a standard flooding algorithm. One way that we can improve energy consumption is through the implementation of a routing strategy that will greatly reduce the number of messages flowing throughout the system. Another main area to focus on in the future is the scalability of our system. Current tests were run on a small network of two piconets, but in a real world situation this number could be in the hundreds or even thousands.

### **6.2.1 Routing**

Routing within the HSNet platform can be implemented in the same way as many types of ad hoc routing. There is a lot of research done into ad hoc routing, and techniques such as AODV, Dynamic Source Routing, and even Delay Tolerant Network (DTN) routing are all areas that could be fruitful to explore.

On top of all of this, HSNet provides other possibilities for routing due to the heterogeneity of the connections in the network. As we've seen in the previous sections of this thesis, different combinations of Wifi Direct and Bluetooth connections yield different throughput values. We have collected a lot of data about throughput in different HSNet scenarios. It would be very interesting to approach routing as a learning problem and weight paths differently based on which links are Bluetooth vs which links are Wifi Direct.

### **6.2.2 Simulation/Scale**

The scale of the experiments run in this work was limited to the actual number of physical devices that we possess. In a more realistic scenario there could be thousands of connected devices. A good area for future work would be to test the scalability of the HSNet platform. It is not feasible to try and get thousands of devices to test on hardware, so looking into simulating HSNet is a good

option to test the scalability. There are a few simulation softwares that might be suitable for this task, including NS3, NetSim, OMNET, OPNET.

I am very interested to see how the throughput scales with a larger network, and I am optimistic that some of the issues regarding the half-duplex nature of Bluetooth and Wifi Direct might be alleviated if there are more nodes in the network to share the load of the messages flowing throughout the system.

## **BIBLIOGRAPHY**

## BIBLIOGRAPHY

- [1] H. Nishiyama, M. Ito, and N. Kato. Relay-by-smartphone: realizing multihop device-to-device communications. *IEEE Communications Magazine*, 52(4):56–65, April 2014.
- [2] Chieh-Jan Mike Liang, Haozhun Jin, Yang Yang, Li Zhang, and Feng Zhao. Crossroads: A framework for developing proximity-based social interactions. In *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*, pages 168–180. Springer, 2013.
- [3] Yufeng Wang, Athanasios V Vasilakos, Qun Jin, and Jianhua Ma. Survey on mobile social networking in proximity (msnp): approaches, challenges and architecture. *Wireless networks*, 20(6):1295–1311, 2014.
- [4] YF Wang, J Tang, Q Jin, and JH Ma. Bwmesh: a multi-hop connectivity framework on android for proximity service. In *Proceeding of The 12th IEEE International Conference on Ubiquitous Intelligence and Computing, UIC*, 2015.
- [5] Farshid Alizadeh-Shabdiz and Suresh Subramaniam. Analytical models for single-hop and multi-hop ad hoc networks. *Mobile Networks and Applications - Special issue: Recent advances in wireless networking*, 11(1):75–90, 2006.
- [6] Preston Gralla. *How wireless works*. Que, 2002.
- [7] Colin Funai, Cristiano Tapparello, and Wendi Heinzelman. Enabling multi-hop ad hoc networks through wifi direct multi-group networking. In *Computing, Networking and Communications (ICNC), 2017 International Conference on*, pages 491–497. IEEE, 2017.
- [8] J. C. Haartsen and S. Mattisson. Bluetooth-a new low-power radio interface providing short-range connectivity. *Proceedings of the IEEE*, 88(10):1651–1661, Oct 2000.
- [9] D. Feng, L. Lu, Y. Yuan-Wu, G. Y. Li, S. Li, and G. Feng. Device-to-device communications in cellular networks. *IEEE Communications Magazine*, 52(4):49–55, April 2014.
- [10] Roy Friedman, Alex Kogan, and Yevgeny Krivolapov. On power and throughput tradeoffs of wifi and bluetooth in smartphones. *IEEE Transactions on Mobile Computing*, 12(7):1363–1376, 2013.
- [11] Saleh Al-Harathi. Half-duplex wireless network scheduling, August 24 2010. US Patent 7,782,803.
- [12] Vaneet Aggarwal and NK Shankaranarayanan. Performance of a random-access wireless network with a mix of full-and half-duplex stations. *IEEE Communications Letters*, 17(11):2200–2203, 2013.
- [13] J Elliott, A Kor, and OA Omotosho. Energy consumption in smartphones: An investigation of battery and energy consumption of media related applications on android smartphones. In *International SEEDS Conference*, December 2017.

- [14] Aaron Carroll, Gernot Heiser, et al. An analysis of power consumption in a smartphone. In *USENIX annual technical conference*, volume 14, pages 21–21. Boston, MA, 2010.
- [15] M. Conti, F. Delmastro, G. Minutiello, and R. Paris. Experimenting opportunistic networks with wifi direct. In *2013 IFIP Wireless Days (WD)*, pages 1–6, Nov 2013.
- [16] Anna-Kaisa Pietiläinen, Earl Oliver, Jason LeBrun, George Varghese, and Christophe Diot. Mobiclique: middleware for mobile social networking. In *Proceedings of the 2nd ACM workshop on Online social networks*, pages 49–54. ACM, 2009.
- [17] Tie Qiu, Ning Chen, Keqiu Li, Daji Qiao, and Zhangjie Fu. Heterogeneous ad hoc networks: Architectures, advances and challenges. *Ad Hoc Networks*, 55:143–152, 2017.
- [18] Hangki Joh and Intae Ryoo. A hybrid wi-fi p2p with bluetooth low energy for optimizing smart device’s communication property. *Peer-to-Peer Networking and Applications*, 8(4):567–577, Jul 2015.
- [19] Trevor Pering, Yuvraj Agarwal, Rajesh Gupta, and Roy Want. Coolspots: Reducing the power consumption of wireless mobile devices with multiple radio interfaces. In *Proceedings of the 4th International Conference on Mobile Systems, Applications and Services, MobiSys ’06*, pages 220–232, New York, NY, USA, 2006. ACM.
- [20] Y. Agarwal, T. Pering, R. Want, and R. Gupta. Switchr: Reducing system power consumption in a multi-client, multi-radio environment. In *2008 12th IEEE International Symposium on Wearable Computers*, pages 99–102, Sept 2008.
- [21] Wenchao Jiang, Zhimeng Yin, Ruofeng Liu, Zhijun Li, Song Min Kim, and Tian He. Bluebee: A 10,000x faster cross-technology communication via phy emulation. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems, SenSys ’17*, pages 3:1–3:13, New York, NY, USA, 2017. ACM.
- [22] Zhijun Li and Tian He. Webee: Physical-layer cross-technology communication via emulation. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 2–14. ACM, 2017.
- [23] Qualcomm. Treppn power profiler, 2016.