OFFLINE-CAPABLE WEATHER APPLICATION USING MULTI-HOP MANET AND Wi-Fi DIRECT

SPECIAL PROBLEM

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

SUBMITTED TO

THE FACULTY OF THE INSTITUTE OF COMPUTER SCIENCE

UNIVERSITY OF THE PHILIPPINES LOS BAÑOS

JAN MAGNUS A. MARIANO

JUNE 2021

ACKNOWLEDGEMENT

To my SP adviser, Ma'am Concepcion L. Khan, who without her guidance I would not have been able to complete this study. I am truly grateful for her advice and suggestions as they gave me a direction in which to proceed with my research.

To all my former lecturers and instructors, who've imparted me with much needed knowledge and allowed me to have a broader understanding of society. Their perspectives not only on academic matters but also in societal topics have expanded my worldview and allowed me to grow as an individual.

To Sigrid France R. Beltran, who continues to support me in all my endeavors as well as accompanying me in all my successes and failures throughout the years.

Lastly, to all my friends and family, for the unwavering support and encouragement, especially during this pandemic wherein the companionship of others is truly precious and needed.

ABSTRACT

Existing weather applications only provide information one-way, but because weather reports can be inaccurate, users of these platforms need a way to be able to report the discrepancies between the weather report and the actual situation in a particular region. The system aims to create a mobile application which will allow users to receive weather data and forecasts as well as send appropriate feedback to the accuracy of the forecast even without the presence of proper network infrastructures. In order to achieve this, a mobile ad hoc network (MANET) utilizing the Wi-Fi Direct technology will be utilized.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

A. Background of the Study

The absence of network coverage for accessing the Internet has always been one of the biggest problems for a majority of the citizens in the country, with reports from DICT logging over 120 million active mobile phone subscriptions during 2016, yet the same report also states that only 34 percent of households have access to the Internet (Department of Information and Communications Technology, 2020). Even more concerning is the presence of only 17,850 cell towers in the country, with a majority of the towers located in the NCR, leaving many rural areas across the country with neither network coverage nor access to the Internet (Caliwan, 2020).

This results in a significant percent of the population living in a digitally isolated state wherein accessing basic data such as political, economic, or meteorological news is difficult. Of particular note is the last point of not having access to meteorological data such as weather forecasts. Such a situation is often costly and even fatal, as not receiving timely reports like upcoming disasters may result in significant economic loss or even risks in one's life due to unpreparedness.

While there are existing technologies such as Automatic Weather Stations (AWS) being used across the country to monitor real time weather data, they are expensive to both set up and maintain. Thus, this study aims to complement such technologies by having mobile ad hoc networks pass data with each other until a central node (in this case, in the form of weather stations) can be reached.

B. Statement of the Problem

Many studies have been conducted towards specifically transmitting data between mobile phones over a large network without existing network infrastructures (Hsieh, 2003; Lin, 2002). However, such studies led to the discovery that long-range communication between mobile devices without external help like custom transmitters is mostly unreliable due to technological limitations at the time.

As such, research towards the topic has been leaning towards intra-network wireless communication such as IoT connections or creation of local ad hoc networks. In contrast, this study aims towards exploring Wi-Fi Direct, a technology described to have Bluetooth's point to point capabilities as well as access range longer than traditional Wi-Fi, with speeds reaching up to 250 Mbps and a range of 200 meters, given clear line of sight with minimal interference (Wi-Fi Alliance, n.d).

Despite these advantages over Bluetooth, developing mobile applications capable of utilizing Wi-Fi Direct for transmitting data has not seen much exploration which include several issues, like the fact that Wi-Fi Direct does not natively support multi-hopping, making ad hoc network implementation difficult. Another prevalent problem is the fact that Android has been decreasing the operations that applications can do without user input.

C. Significance of the Study

This study will provide a way for individuals living in hard to reach areas without any network access to be able to receive updated weather data, with the limitation being that a Wi-Fi

Direct capable device is within detection range. Another benefit that this work can bring is a way to crowdsource weather forecast feedback, as end-users can potentially send responses on the accuracy of weather forecasts even without being connected to the Internet.

While the resulting feedback is limited to a simple confirmation whether the forecast is correct or not, the responses might help in predicting future weather conditions over a finer region of interest, especially because there are only a limited number of weather stations around the country. Lastly, this study experiments with a different routing protocol, namely Ad-hoc On-demand Distance Vector (AODV) (Belding-Royer, 2003), for employing Wi-Fi Direct in mobile ad hoc networks, as opposed to previous works, which will be explored later.

D. Objectives of the Study

This research aims to achieve the following:

1. Collect weather forecast and weather information data from the OpenWeatherMap API and disseminate information from the server to end-users.
2. Develop a weather monitoring mobile application that allows end-users to send feedback regarding the accuracy of the weather forecast, as well as be able to provide proof of their feedback through captured images or videos.
3. Allow said mobile application to receive or send data without any network access (internet connection and mobile network signal).
4. Store and organize user responses for future use.

E. Scope and Limitations of the Study

The ad hoc operations involving the Wi-Fi Direct technology is not intended for practical usage due to inconsistencies with the Android API library and Android source code. The mobile application will also only be tested for devices running Android 6.0 until Android 10, as other versions outside this range either do not support Wi-Fi Direct, or involve core changes in their security that do not allow for the study to perform implicit commands during the application's run time.

CHAPTER II

REVIEW OF RELATED LITERATURE

A. System Architecture

This work is inspired by a study conducted by Arroyo-Tandang (n.d.) entitled "Data Synchronization and User Targeting based on Segmented Cross-Platform Push Notification". The study explored the option of utilizing cloud services, namely Firebase Cloud Messaging, coupled with content filtering in order to minimize data consumption in the delivery of data to a mobile application. Arroyo-Tandang's system architecture will be adapted and modified in order to reduce overhead during development since both studies are geared towards offloading the processing of data to the server instead of the client. This work will also entail a weather application that uses multi-hopping ad hoc networks.

B. Multi-hop Ad Hoc Networks

Ad hoc networks are defined as temporary, often wireless, networks wherein devices can communicate between each other without a wired infrastructure to facilitate the network. This is in contrast with the more conventional network set up wherein a central hub keeps track of the information of the connected devices in order to handle the transmission of data between nodes in the network as well as handle communication with other networks. An example of such is a home network wherein the router serves as the central hub and is connected physically through wires with the ISP.

Because there is no central point, there is a need for a device to know what path its data should take when it wants to send information to another node within the network, especially if

the destination node is outside of its immediate range. To achieve this, a *routing protocol* is employed. A routing protocol is a mechanism that allows a node in the network to discover an optimized path to a destination node, with said path often referred to as a *route*.

After a suitable route has been found, the source device transmits the data to the first neighbor along the path, where the receiving neighbor then sends the data to the next target and so on. This process is called multi-hopping, and is often needed for mobile ad hoc networks (MANET) because devices in MANETs have limited scanning range and can only repeatedly offload their data over short distances.

C. MANET Routing Protocol

Before discussing the routing protocols explored, there is a need to describe the characteristics of a MANET (Alaqel, 2014). Firstly, MANETs have dynamic network topologies, meaning devices can enter or exit the network freely. Secondly, devices within the network often have limited bandwidth capabilities. Lastly, devices within a MANET rely on some form of battery. With the above characteristics taken into consideration, routing protocols for mobile ad hoc networks should have a robust structure to adapt with frequent changes as well as minimal resource consumption.

Moving on, there are two major categories of routing protocols for MANETs, with the third one being hybrid routing protocols which will not be discussed because it is a mix of the first two. The first category is called *proactive protocol* (Maan & Mazhar, 2011). Proactive protocols allow nodes to consistently maintain a complete topology of the network, making the

delay in transmission of data low since the device has knowledge of all routes in the network. Another advantage of proactive protocols include high network availability as the device is always updated to changes in the network. Disadvantages for proactive protocols include high battery usage as well as high demand for storage since the device keeps track of routing information for all the devices in the network.

The second category is called *reactive protocol*, and functions by performing the route discovery process only when needed. The advantages of reactive protocols include lower energy consumption since route finding operations are only performed whenever needed. Reactive protocols are also less of a burden on the device since it does not store the routes for all the nodes in the network. Disadvantages include the high delay for sending and receiving packets between devices, as each node in the network has to flood the network with broadcast packets to help the source node in finding a valid route.

D. Ad Hoc On-Demand Distance Vector Routing

This study will utilize a modified version of the Ad Hoc On-Demand Distance Vector (AODV) routing protocol. AODV works by having the source device broadcast a route request (RREQ) packet to its neighbors. The RREQ packet contains the IP address of both source and destination nodes, sequence number which serves as a pseudo-identifier for a device, and hop count, which iterates every time a packet travels from one device to another during the route discovery process.

After the source device broadcasts its RREQ, a neighboring device (with no previous route information) then stores the IP address and sequence number of the source node, so that the neighboring device can form a link with the source node. The neighboring device also sets an arbitrary amount of time as the lifetime of a link towards any node, so that old links are eventually discarded when they are not used to reduce overhead and prevent accessing of nodes that are inactive.

When the next neighboring device receives the RREQ, it also records the previous path taken, so the neighboring device now has knowledge of the path to the previous hop as well as the source node. When the RREQ reaches the destination node, the destination node then ends up with a complete path due to the information stored by all other previous nodes to the source node. The destination node then sends a route reply (RREP) packet which contains the destination and source node's IP addresses, the destination node's own sequence number, and the total hop count. A unicast connection is then made from the destination node to the previous node that sent the RREQ to the destination node, instead of broadcasting the RREP, so as to avoid even more flooding of packets in the network.

Once again, the nodes lying in the path between the source and destination nodes will receive information about the destination node, and thus have a complete route from source to destination when needed, given that the links are all still alive due to the lifetime restriction. Alternatively, if a neighboring device has a route to the destination node that is still valid according to its lifetime, it can send a route reply (RREP) packet to the source node, on behalf of

the destination node. When the RREP is received by the original source node, only then will the transmission of data begin since a route has been found.

E. Wi-Fi Direct Integration

Wi-Fi Direct is a recent standard by the Wi-Fi alliance that allows devices to establish a connection with each other without the need for an access point. As a competitor of the Bluetooth technology, Wi-Fi Direct offers a maximum range (given clear line of sight) of up to 200 meters and speeds up to 250 Mbps. By embedding a SoftAP (software enabled access point) into devices that support it, single-hop communication between connected devices can be performed. This allows two devices that are connected to different Wi-Fi networks to share data between each other without disconnecting to their respective networks, as opposed to creating mobile hotspots.

To discuss its disadvantages, one of the biggest drawbacks towards exploring the viability of the technology is the fact that it is built for point to point communication between two devices, and multi-hop communication is not natively supported. This is because when connections between two devices are formed, a host/Group Owner (GO) and client role is determined between the two, and neither of the devices are able to be part of another Wi-Fi Direct network while they are still connected (See Figure 1).

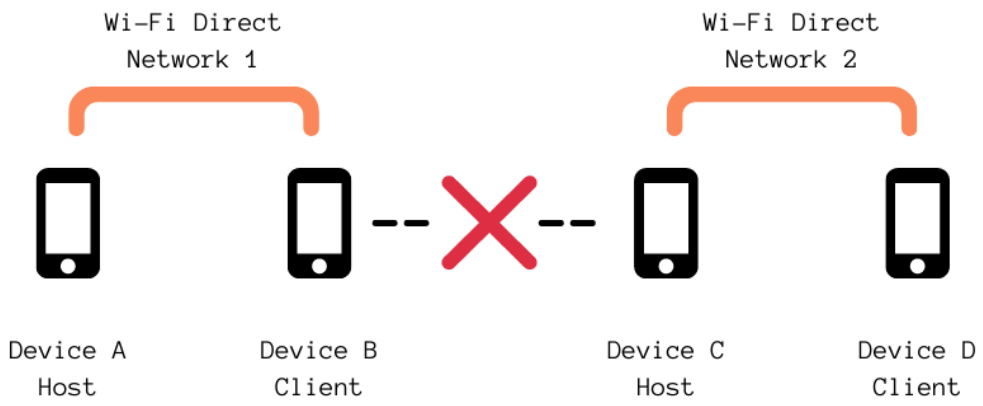

Figure 1 : Device B cannot communicate with Device C

Several workarounds for making a multi-hop MANET that uses Wi-Fi Direct have been explored, with one of the earlier ones being a study by Casetti et al. (2015), which bypasses the fact that a device cannot be both a GO and client by having the host of a Wi-Fi Direct network connect as a legacy client to the host of another network. This is possible because the Android API allows a GO to expose its network as a traditional access point, allowing devices not supporting Wi-Fi Direct to connect into the network through the Wi-Fi interface.

Because a complete network is formed from the start, no routing protocol is used and instead, broadcast packets containing the data to be transmitted are used over UDP protocol until the desired destination is reached. One disadvantage of the presented solution is that devices are perpetually connected to the network, thus battery usage remains high even when the node does not need to transmit data along the network. Also, when a node leaves the network, fixing the network becomes difficult because routing information has to be flooded across the whole network so all devices can maintain an updated topology. Another factor that makes this

approach hard to implement is that when a GO creates its Wi-Fi Direct network, it takes a static IP address of 192.168.49.1, and thus, addressing IP address conflicts introduces another layer of complexity to the problem (Xu, 2017).

A more recent study looked into using single-hops and the Destination-Sequence Distance Vector routing protocol (Ahmed & Ali, 2020) to effectively transmit data from one client through the other. The data transmission starts by having the source device discover nearby devices in listening mode and connect to the nearby devices in a 1 to 1 manner in order to exchange information about other discovered devices in the vicinity of each device. When the source node finishes exchanging information to all nearby devices, it gains a complete map of the network and can send data by connecting to the nearest node along the path and when data transfer is complete, the connection is terminated to free up both devices for future links with other devices. The node that receives the data from the source node then does the same to the next node along the path, connecting and disconnecting when data transmission is finished, until the data sent by the source node is received by the intended destination node.

A similar approach will be used in this study but instead of DSDV, AODV will be used and routing information will be exchanged by broadcasting through the network instead of initiating 1 to 1 connections.

CHAPTER III

METHODOLOGY

A. System Design Overview

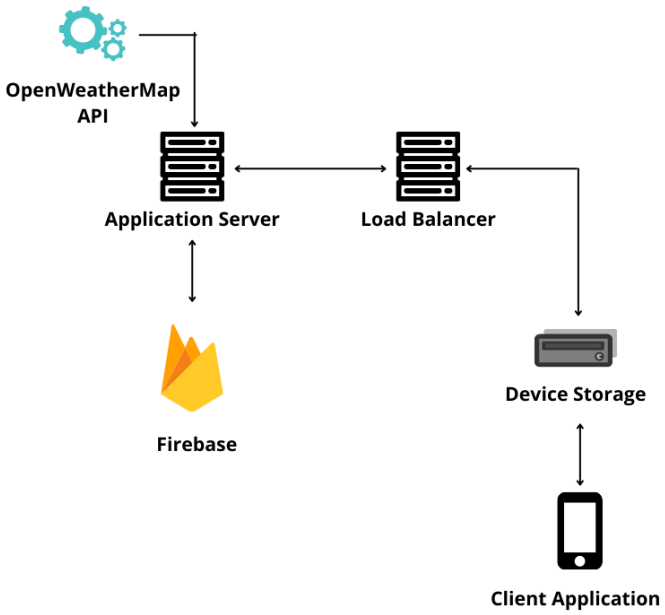

Figure 2 : System Design

The application server will function as the backbone of the system and is responsible for a majority of the data management that will occur. Firstly, the application server will accommodate all API calls that will be made. Because of the limitations in the number of calls that can be made with the API used in this study, the system will run a task scheduler that fires API calls periodically during a set time of the day in order to collect the weather data from the roughly 1600 municipalities in the Philippines. Secondly, the application server is the only part of the system that can communicate with the databases, and all interactions will be reflected in the application server. Thirdly, any data that needs to be processed such as raw weather data from

the OpenWeatherMap API and user feedback including files sent will be handled by the application server.

The web server will be responsible for handling all HTTP requests from the clients, especially with relaying POST requests to the application server. The web server will utilize JSON web tokens (JWT) to perform user authentication operations and also track session data in order to prevent data collisions.

B. Mobile Application

The mobile application will serve the following functions :

1. Deliver weather data to the end-user, filtered by their region of interest.
2. Allow users to give feedback on weather reports using predetermined responses, and also allow users to attach images or videos for further proof of their feedback.
3. Allow users to view the reports of other users, filtered to the chosen area of interest.
4. Allow users to receive weather data or send feedback data to the server even without an internet connection through the usage of a mobile ad hoc network.

On top of the presented functions, the mobile application will also employ caching to reduce the user's bandwidth consumption while connected to the Internet. Such data-saving practices include having the application server return data from GET requests only when the specified resource has changed, as well as caching images from received user reports. User

sessions are also preserved by having the device refresh the application's access token to the server to avoid having the user enter their credentials when the application is closed.

C. Offline capability and Android API Limitations

In order for the application to send or receive data without internet connection, a modified version of the AODV routing protocol is used. For this, the Android API's implementation of Network Service Discovery (NSD) using the Wi-Fi Direct library will be used during the route discovery process. This is because NSD allows broadcasting of small packets of data to devices within Wi-Fi range. The following section briefly describes the modified AODV used in this study.

Initially, each node in the network initializes two values, the device's own MAC address and its \textit{sequence number}. The sequence number is used to uniquely identify a device's state in order to prevent potentially stale information from being stored and also avoid looping routes. This study uses the device's MAC address instead of IP address because Wi-Fi Direct connection utilizes MAC addresses to form connections between devices.

The device will also initialize a routing table, which will store all discovered routes from the route discovery process. An entry in the routing table contains the following information : Destination device MAC address, Destination device sequence number, Next hop MAC address, the MAC address of the nearest device to be used as a "hop" to reach the destination, \textit{hasConn} flag, used for checking if a destination has access to the Internet, \textit{hop count}, used for telling how many hops, or how many more devices need to be passed through,

before the destination, and *lifetime*, an arbitrary value set upon the discovery of a route, which helps keep routes up to date by discarding unused routes.

The modified route discovery starts with the source device, assumed to have no entries in its routing table, increment its sequence number and broadcast an RREQ packet containing its own MAC address, sequence number, and a hop count, which is initialized at 1. Other parts of the RREQ include Next hop MAC address, which will be empty since there is no previous hop, destination MAC address, also empty because there are no entries yet in the routing table to serve as destinations, and destination sequence number. The RREQ is broadcast multiple times over a period of time, labeled as *BCAST_TIME*. Every time the *BCAST_TIME* is reached, the RREQ is broadcast again but with the sequence number being incremented and with the *BCAST_TIME* being twice as long as the previous one, up until the *MAX_RETRIES* is reached. *MAX_RETRIES* is an arbitrary value to decide how many times the device should try broadcasting.

Any device that receives the packet then stores the MAC address, sequence number, and current hop count from the RREQ along with a *lifetime* value. The receiving device also checks if it is the intended destination by checking if it has access to the Internet. From here, the receiving device has three actions to take : Rebroadcast the RREQ if it has no access to the Internet, Send an RREP back to the source device, or Send an RREP if it has a route to a device with access to the Internet and also with said route not being stale according to its *lifetime*.

If the receiving device rebroadcasts the RREQ, it appends its own MAC address to the RREQ, and increments the hop count by 1. The receiver's MAC address is added to allow the next receiving device to know what device it should connect to in order to reach the original source node.

If the receiving device has access to the Internet, it then increments its own sequence number and sends an RREP back to the previous node, to be delivered all the way to the source node. The RREP contains the destination node's MAC address, sequence number, a fresh hop count because the RREP might discover a shorter path along the way, the source node's MAC address, and next hop MAC address, which will be null because there is no other hops made yet in the RREP delivery process. Instead of unicasting the RREP to the previous node as per the original AODV algorithm, the RREP will be sent using broadcasting because the Wi-Fi Direct API does not present any methods for unicasting other than creating a connection, which is an expensive process.

After a valid route has been found, the source device then sends a connection request through the Wi-Fi Direct interface to the next hop along the route. Figure 3 presents a simple example of the route discovery process using three devices.

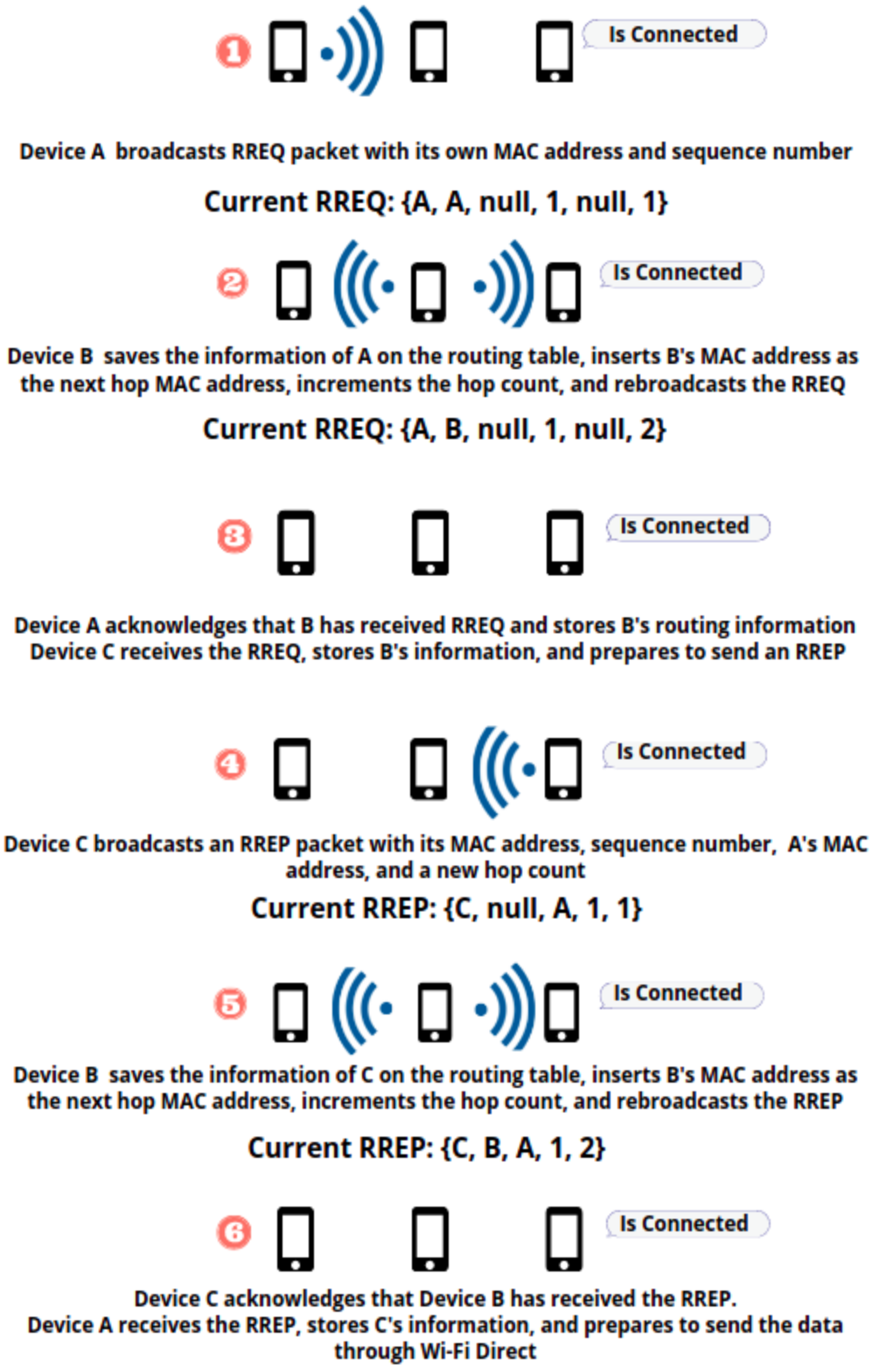

Figure 3 : AODV Route Discovery

It must be noted that the NSD implementation for Android remains unreliable due to issues on the Android source code itself as well as differences on how various Android versions handle the execution for broadcasting packets. Such problems include sudden unpredictable periods wherein the service discovery stops receiving broadcast packets. A fix that raises the consistency of the discovery process is by restarting the Wi-Fi interface, which cannot be done programmatically on newer versions of Android (Android 10 and above) and requires the user to switch the Wi-Fi on and off manually, which can be detrimental to user experience.

Another UX-related problem caused by a limitation on the Android API is that when a device requests another device through the Wi-Fi Direct interface, the invited user needs to respond via a button press within a set time limit according to the WPS configuration. This feature is present on all devices that support Wi-Fi Direct and can only be removed by having root privileges, which this study will not cover. An alternative solution to this UX problem is by using the legacy Wi-Fi mode of the Wi-Fi Direct interface, but this approach will not be implemented because of energy consumption concerns.

CHAPTER IV

RESULTS AND DISCUSSION

The resulting mobile application was tested on three devices, with the following details :

1. One Samsung A20 device with API 29 (Device A)
2. One Samsung J7 device with API 23 (Device B)
3. One Samsung A01 device with API 29 (Device C)

The test was conducted in an outside environment wherein devices A, B, and C are placed with equal distance to each other such that device A will not be able to directly communicate with device C.

The testing procedure comprised 5 runs with 2 input variables, the first being the distance between each device, consisting of 30 meters per device (total of 60 meters from device A to C), 40 meters per device, 50 meters per device, and 60 meters per device. The second input variable was the payload transmitted over Java socket, with the payloads being divided into a ping packet, a 1 MB byte array, a 5 MB byte array, and a 10 MB array. With 5 runs per distance and per payload, the testing phase comprises 100 data exchanges, starting from route discovery until the destination receives the payload.

The test primarily looked at finding the effective limit of Wi-Fi Direct transmission given different distances as well as transmitting with different sizes of payload. Figure 4 and the tables below show a summary of the results.

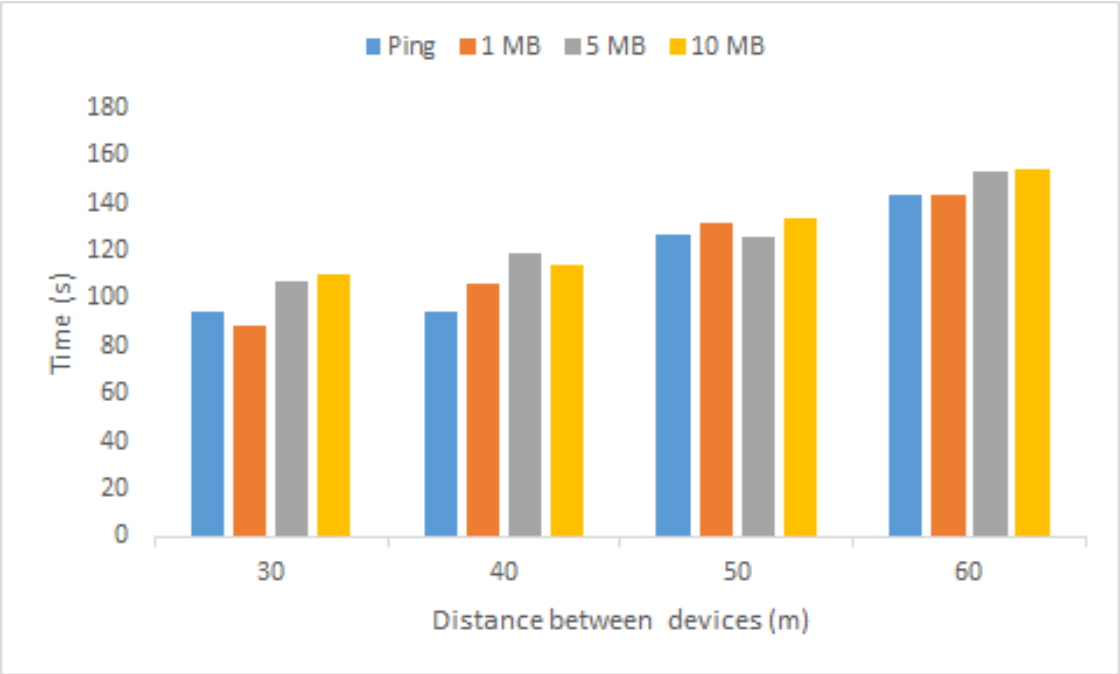

Figure 4 : 2-hop data transfer by distance and payload

	RREQ Travel	RREP Travel	Data Transfer	Total
Ping	52.77s	40.13s	1.50s	94.40s
1 MB	46.64s	39.88s	2.24s	88.76s
5 MB	61.36s	41.89s	3.89s	107.14s
10 MB	62.47s	41.24s	5.98s	109.70s

Table 1 : 30 meters distance between each device

	RREQ Travel	RREP Travel	Data Transfer	Total
Ping	48.51s	44.26s	1.73s	94.51s
1 MB	61.05s	41.63s	3.59s	106.28s
5 MB	67.95s	45.75s	5.47s	119.17s
10 MB	60.81s	45.85s	7.42s	114.09s

Table 2 : 40 meters distance between each device

	RREQ Travel	RREP Travel	Data Transfer	Total
Ping	67.27s	57.68s	2.27s	127.22s
1 MB	71.38s	55.76s	4.10s	131.23s
5 MB	68.38s	51.09s	6.21s	125.67s
10 MB	65.48s	60.18s	7.98s	133.64s

Table 3 : 50 meters distance between each device

	RREQ Travel	RREP Travel	Data Transfer	Total
Ping	77.21s	63.63s	2.84s	143.67s
1 MB	70.81s	67.58s	5.17s	143.57s
5 MB	79.85s	65.84s	7.75s	153.45s
10 MB	81.77s	63.40s	9.09s	154.26s

Table 4 : 60 meters distance between each device

Figure 4 and tables 1 to 4 show a consistent increase in the total time as both payload and distance increases. However, one thing to note is how the route discovery process (Route Request and Route Reply) fluctuates randomly. This is because the average discovery time gets heavily skewed whenever the NSD process fails, prompting the application to restart the broadcasting. Device A which runs Android API 23 seems to be the most susceptible to this behavior, with the 2 devices running Android API 29 exhibiting lower occurrences of NSD failures. Furthermore, speed of data transfer remains around the range of 25 to 50 Mbps, way below the proposed 250 Mbps of the Wi-Fi Direct standard.

Next, packet loss during data transmission with different distances and payloads. For this test, 1000 ping packets are exchanged between two devices connected with Wi-Fi Direct and Java sockets. Unique IDs are assigned to each packet and are logged by the receiving device, to

analyze which packets did not arrive. The distances between each device consisted of 10 meter increments from 30 to 100 meters, and the devices used are Device B and C from the previous payload test, both running Android API 29 to reduce other variables that might alter the data.

	Received Packets	Failed Packets	Packet Loss (%)
30 meters	996	4	0.4
40 meters	995	5	0.5
50 meters	991	9	0.9
60 meters	950	50	5.0
70 meters	907	93	9.3
80 meters	848	152	15.2
90 meters	812	182	18.2
100 meters	754	246	24.6

Table 5 : Packet Loss Percentage

The packet loss table shows that Wi-Fi Direct connections perform well at distances less than 60 meters and experience heavy packet loss at a range from 60 to 70 meters, and at distances more than 80 meters, the packet loss ratio spikes up once again. At distances more than 100 meters, Wi-Fi Direct networks experience disconnections between host and client, rendering them unusable for practical use as transfers can take several retries.

CHAPTER V

CONCLUSION

In conclusion, the study explored the implementation of an infrastructure-less network to transfer data across a large distance using Wi-Fi Direct and AODV routing protocol. The experimental results showed that a MANET built with Wi-Fi Direct has sufficient throughput to support the transferring of large files. Another benefit of the large transfer rate is the fact that flooding techniques can be used for route discovery as there is no need to worry about data usage. However, one major flaw of the Wi-Fi Direct technology in regards to being used for MANETs is its inability to natively support transmitting of small packets without an established connection, leading to a situation wherein Wi-Fi Direct networks are only limited to the Group Owner's own detection range. Because of this, workarounds such as using broadcasting techniques or individually establishing connections with devices within scanning range are adapted, and these solutions often are sub optimal, as shown by the slow route discovery process of this study.

CHAPTER VI

RECOMMENDATION

In order to enhance the practical use of the application, future studies can integrate the usage of long range point-to-point receivers or other signal extenders since the Wi-Fi Direct technology can include other devices in its network with minimal set up. An alternative technology that can be explored in the future is Bluetooth 5.0, as the Android API for Wi-Fi Direct is not yet stable, especially across different device manufacturers.

BIBLIOGRAPHY

Ahmed, I. M., & Ali, H. M. (2020). Building a Dynamic Multi-hop WiFi Direct Network For Android Smartphones. Solid State Technology, 63(2s).

Alaqel, H. A., Alslaim, M. N., & Zaghloul, S. S. (2014, April). A comparative study of MANET routing protocols. In The Third International Conference on e-Technologies and Networks for Development (ICeND2014) (pp. 178-182). IEEE.

Arroyo-Tandang, J. (n.d.). Data Synchronization and User Targeting Based on Segmented Cross-Platform Push Notification. University of the Philippines Los Baños

Belding-Royer, E., Das, S., & Perkins, C. (2003). RFC3561: Ad hoc on-demand distance vector (AODV) routing.

Caliwan, C. L. (2020, December 1). LGUs approve 2.2k telco tower permits: DILG. Retrieved from https://www.pna.gov.ph/articles/1123445.

Casetti, C., Chiasserini, C. F., Del Valle, C., Duan, Y., Giaccone, P., & Pelle, L. (2015, June). Content-centric routing in Wi-Fi direct multi-group networks. In 2015 IEEE 16th international symposium on a world of wireless, mobile and multimedia networks (WoWMoM) (pp. 1-9). IEEE.

Department of Information and Communications Technology. (2020, June). NTC data as of June 2020 (internal copy). Retrieved from https://dict.gov.ph/ictstatistics/wp-content/uploads/2021/03/NTC-data-as-of-November-2020.pdf

Hsieh, T. Y., Hsu, C. S., & Tseng, Y. C. (2003). Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks. Computer Networks, 43(3), 317-337.

Lin, C. Y., Sheu, J. P., Tseng, Y. C., & Wu, S. L. (2002). A multi-channel MAC protocol with power control for multi-hop mobile ad hoc networks. The computer journal, 45(1), 101-110.

Maan, F., & Mazhar, N. (2011, June). MANET routing protocols vs mobility models: A performance evaluation. In 2011 Third international conference on ubiquitous and future networks (ICUFN) (pp. 179-184). IEEE.

Wi-Fi Alliance. (n.d.). Wi-Fi Direct. Retrieved from https://www.wi-fi.org/discover-wi-fi/wi-fi-Direct

Xu, J. (2017). Wi-Fi Direct Multi-Group Communication: Connect different Wi-Fi Direct groups with Access Point (Doctoral dissertation, The Ohio State University).