

Master Project Report

Seamless Handover among WiFi Access Points on Raspberry Pi

Project Advisor: Prof. Raj Jain

Author: Yijian Li, Email: liyijiankaka@gmail.com

Abstract

The main goal of this project is to realize seamless handover among cellular stations in a cellular system for emergency and disaster relief. Raspberry Pi with OpenWRT is chosen to become WiFi Access Point as the cellular station. Its limited weight, size and energy consumption enable it to be functional on a micro unmanned air vehicles (UAV). The overall design for WiFi seamless handover is the Wireless Distribution System (WDS) mode which is supported by several open source WiFi solutions including OpenWRT. The result is the switch over time within the same network is less than 0.3s. The phone call via Voice over IP (VoIP) software are not dropped during the handover.

Keywords: seamless handover, WiFi, Raspberry Pi, cellular system, OpenWRT, unmanned air vehicles, UAV, Wireless Distribution System, WDS, Voice over IP, VoIP, Access Point.

Table of Contents

1. Introduction.....	2
2. Project Background.....	2
2.1 Introduction to the Overall Research Project.....	2
2.2 Seamless Handover's Role and the Existing Work.....	3
3. The WiFi seamless handover design.....	3
3.1 Wireless Distribution System (WDS).....	3
3.2 OpenWRT and its WDS mode on Raspberry Pi.....	4
3.3 WDS Implementations Across Different Platforms.....	4
4. The implementation and verification of the design.....	4
4.1 The implementation of OpenWRT on Raspberry Pi.....	4
4.2 The implementation of WDS mode network.....	6
4.3 The verification of the seamless handover function.....	9
5. Field Test of the Design.....	11
5.1 Coverage and Signal Strength.....	11
5.2 Network speed.....	12
5.3 Switch over time.....	13

6. Future Work	13
7. Summary	14
8. Acronyms	14
9. References.....	14

1.Introduction

In this project, I have successfully implemented seamless handover function in the context of a UAV based cellular station network. The design is primarily based on the Wireless Distribution System supported by open source WiFi software. The WiFi Access Point on the UAV is OpenWRT on a Raspberry Pi.

Section 2 gives the detailed background introduction. The overall design is in Section 3. Section 4 shows the actual implementation and the verification of the seamless handover function. More field test results are provided in Section 5.

2. Project Background

The communication system in today's world is very fragile facing large scale nature disasters as it relies on numerous infrastructures and power supplies. The communication service interruption will cause many severe problems. The micro UAVs provides a new solution as it can be instant cellular tower in the sky anytime and anywhere with very limited resources.

2.1 Introduction to the Overall Research Project

The overall research project is to design and implement a wireless network based on the use of micro UAVs to form a reliable cellular system for emergency and disaster relief. The goal is to integrate the latest networking technologies to enhance the performance of the system in the unpredictable disaster environment.

WiFi is one of the wireless technologies for cellular stations on UAVs. It is the most widely used wireless communication solution while VoIP is an important form of phone service nowadays. Furthermore several powerful open source WiFi software, such as OpenWRT are available. This is helpful to maintain the budget for this low-cost system. All the routers and adapter used in this project are WiFi products. 3G and 4G will be implemented in future projects.

So far we have done the proof of the concept and some research on the devices available in the market. Two working UAVs with commercial routers have been tested, including dozens of test flights in open areas. We can easily replace the commercial routers with Raspberry Pis as they can share the same power supply and have similar weight and size. There are other candidate routers available and we may conduct more tests on them to determine which one delivers the best performance.

2.2 Seamless Handover's Role and the Existing Work

Seamless handover is one of the key requirements for the project. The frequent replacement of the UAVs due to the small battery volume onboard will cause unacceptable service interruption if seamless handover is not there. In disaster environment, every minute counts. Redialing may be too expensive for those who need instant help.

Seamless handover had not been demonstrated before this project. However several possible WiFi solutions on Raspberry Pi have been implemented and tested. Also multiple kinds of WiFi USB adapters have been tried.

Raspberry Pi has been popular in recent years. People have managed to use it as a WiFi router in different ways. One of them is the open source WiFi project OpenWRT [RPi1]. On the other hand, WDS mode has been available since 2007 and it is supported by most chip makers. The seamless handover function of WDS mode has been verified on several commercial routers. However no one has either tried WDS mode or seamless handover function on Raspberry Pi. My work is mainly to implement this new combination, i.e. Raspberry Pi + OpenWRT + WDS.

3. The WiFi seamless handover design

In WDS mode, the Raspberry Pi equipped with OpenWRT works as the repeater of the base router. All the clients connected to any Raspberry Pi share the exact same network information including Dynamic Host Configuration Protocol (DHCP) pool. In this way, the seamless handover becomes possible as clients always think they are still connected to the same WiFi network.

3.1 Wireless Distribution System (WDS)

WDS has been available for years and supported by most commercial chips. However it has never been standardized. Therefore, compatibility issue is one of the major problems when trying to implement it among different platforms, even those from the same vendor.

There are two modes of WDS, wireless bridging and wireless repeating [wiki1]. We are using wireless repeating in this project because it allows the wireless clients to connect to the stations and communicate with each other. WDS has two major disadvantages. First, the maximum wireless effective throughput may be limited. We can see the negative effect on the network throughput in Section 5 where the field test results will be shown. This issue is caused by the fact WiFi is an inherently half duplex medium and, therefore, any WiFi device functioning as a repeater must use the store and forward method of communication [wiki1]. This limits the throughput significantly. Second, complicated encryption scheme may not work on the network. This is not an issue in our project because in emergency relief environment, it is not possible to let the clients have any kind of password in advance. This is also why we

did not implement any form of encryption in the system.

3.2 OpenWRT and its WDS mode on Raspberry Pi

OpenWRT is an open source router firmware. It has official support for many commercial routers and numerous related projects and application are available online. The Broadcom 2708 chip used in Raspberry Pi is supported by OpenWRT [RPi1]. However the official image has some files missing which leads to booting issue. The the solution and the required booting files are available on a different site [RPi2]. After successful installation, one can configure it via Ethernet or wireless connection as long as both are in the same subnet.

The original OpenWRT image does not support USB WiFi adapters. It is necessary to install some WiFi related software and the drivers for the particular USB WiFi adapters being used. After this step, you will be able to access the WiFi configuration page and it can be used as a wireless router. There are several other necessary steps which will be explained in detail in Section 4.

In OpenWRT, WDS is a non-standard extension to the client mode [WDS1]. It can be easily set up in the WiFi configuration page. If the configuration is correct, the association between the WDS access point and the WDS client will be automatically done. With good signal strength, the WDS connection is very stable and works just like any normal WiFi connection.

3.3 WDS Implementations Across Different Platforms

Although WDS is not standardized, it can be used among different platforms. In this project, the base router is a Linksys WRT54G v5 router with DD-WRT firmware which is another well-known third party firmware. It is compatible with Raspberry Pi's WDS implementation.

The compatibility issues require that any possible combination of different platforms used in the project be tested in the field in order to determine whether it is a valid combination or not. It is not an ideal situation as we may not be able use WDS on the combination which produce the best performance in terms of coverage and signal strength. We have tried several other commercial WiFi routers but ended up putting them on hold because of the WDS connection compatibility issue.

4. The implementation and verification of the design

In this Section, we describe the actual implementation of the design. This includes how to implement OpenWRT on Raspberry Pi, how to enable WiFi function of the OpenWRT and the detailed configuration of the WDS mode. Finally we show the verification of the seamless handover function of the design.

4.1 The implementation of OpenWRT on Raspberry Pi.

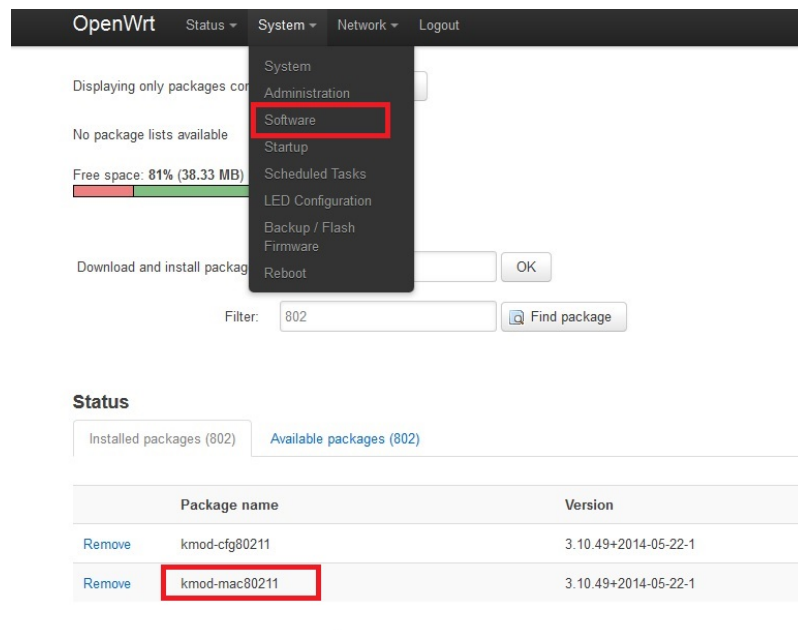
The necessary image files are available at [RPi1]. Download the image file and write it to the SD card you will use to boot the Raspberry Pi. Then we need some additional booting files which are provided in [RPi2]. Replace the original files with the new ones. After this, the OpenWRT will be up when the Raspberry Pi with this SD card has proper power supply, either battery or USB power supply.

For the initial configuration, you need to connect any computer to the Raspberry Pi via Ethernet cable. There are several ways to figure out the IP address of the Raspberry Pi. Once the IP address is obtained, use SSH connection to access it to set up the password for the admin. Then all the remaining configuration can be done in any web browser as OpenWRT uses LuCI as its web graphical user interface (WebGUI) [LUCI].

First the LAN port is configured as a DHCP client with a fixed IP address to make future access and configuration easier. Next install the WiFi related software packages to enable wireless function of the OpenWRT. This can be done easily using the OPKG Package Manager [OPKG]. Also the driver for the USB adapters, Alfa AWUS036NHR v2 in this project, are required:

1. kmod-mac80211: the WiFi package;
2. kmod-rtl8192c-common: the driver for the adapters;
3. wpad: for potential encryption configuration

Figure 1 shows how to find a software and Figure 2 shows the WiFi configuration page after successful installation.



The screenshot shows the OpenWRT LuCI web interface. At the top, there is a navigation bar with 'OpenWrt', 'Status', 'System', 'Network', and 'Logout'. A dropdown menu is open under 'System', with 'Software' highlighted in red. Below the menu, there is a search bar with '802' entered and a 'Find package' button. The main content area shows 'Status' with 'Available packages (802)' selected. A table lists installed packages:

	Package name	Version
Remove	kmod-cfg80211	3.10.49+2014-05-22-1
Remove	kmod-mac80211	3.10.49+2014-05-22-1

Figure 1. Install the necessary packages for OpenWRT

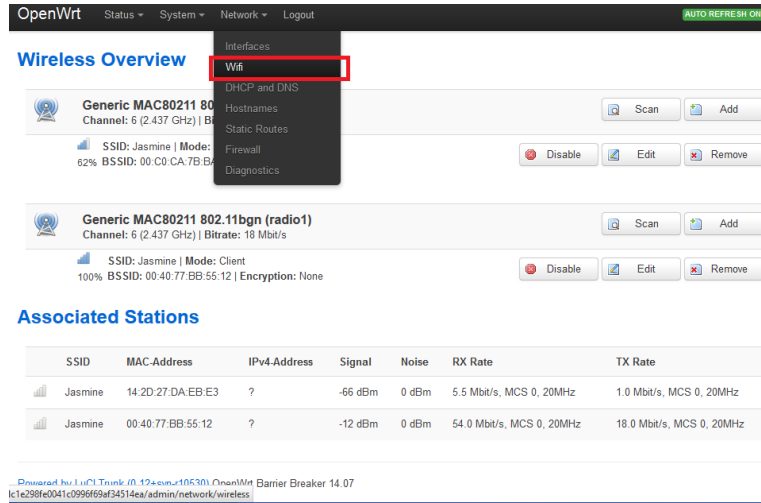


Figure 2. WiFi Configuration page

Reboot the Raspberry Pi and the WiFi configuration page can be accessed. Add a new interface for the wireless network and set the firewall properly. The Raspberry Pi is a WiFi router now. When a USB WiFi adapter is connected via a USB port, it will appear as a new device in the WiFi configuration page. Two adapters are needed, one for WDS connection and one for access point. The figures of wireless configuration will be shown later in this section.

4.2 The implementation of WDS mode network

First the base router, which is Linksys WRT54G v5 in this project, is configured as the WDS access point. DD-WRT is used on this router and it can be accessed via web browser. Add the MAC address of the very WiFi adapters used for WDS connection to the WDS client list. As illustrated in figure 3, several MAC addresses have been added to the list.

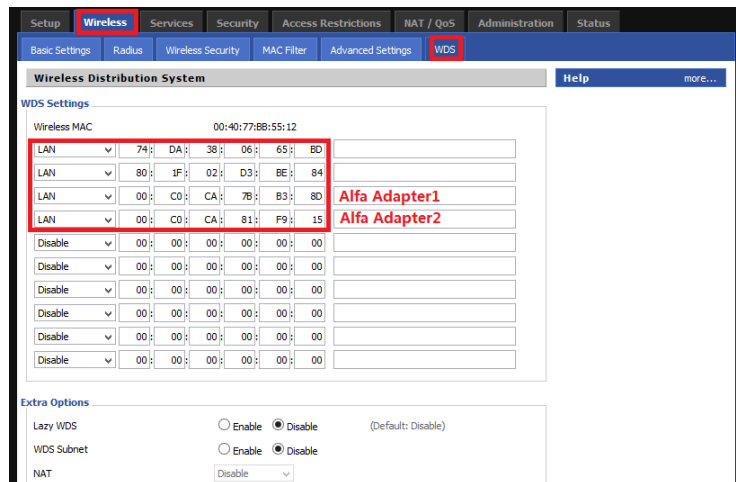


Figure 3. Base router WDS setting

Then the OpenWRT is configured as the WDS client. The steps are simple. However the following instructions are important:

1. WDS client must have the same Service set identification (SSID), encryption method, password and WiFi channel as the base router.
 2. The DHCP server on the WDS client must be turned off. Any wireless client connected to the WDS client share the same DHCP pool on the base router.
 3. The two USB adapters share the same wireless interface.
- The figures below show the actual configuration used in the project.

The screenshot shows the OpenWRT web interface. At the top, there is a navigation bar with 'OpenWrt', 'Status', 'System', 'Network', and 'Logout' menus, and an 'AUTO REFRESH ON' button. Below this is the 'Wireless Overview' section. It contains two radio interface cards:

- radio0 (Access Point):** Channel: 6 (2.437 GHz), Bitrate: 1 Mbit/s. SSID: Jasmine, Mode: Master, Encryption: None. BSSID: 00:C0:CA:7B:BA:05.
- radio1 (WDS client):** Channel: 6 (2.437 GHz), Bitrate: 18 Mbit/s. SSID: Jasmine, Mode: Client, Encryption: None. BSSID: 00:40:77:BB:55:12. A red box highlights the BSSID and a note below it says 'MAC address of the base router'.

Below the radio cards is the 'Associated Stations' table:

SSID	MAC-Address	IPv4-Address	Signal	Noise	RX Rate	TX Rate
Jasmine	14:2D:27:DA:EB:E3	?	-62 dBm	0 dBm	1.0 Mbit/s, MCS 0, 20MHz	1.0 Mbit/s, MCS 0, 20MHz
Jasmine	00:40:77:BB:55:12	?	-4 dBm	0 dBm	54.0 Mbit/s, MCS 0, 20MHz	18.0 Mbit/s, MCS 0, 20MHz

At the bottom, it says 'Powered by LuCI Trunk (0.12+svn-r10530) OpenWrt Barrier Breaker 14.07'.

Figure 4. The WiFi configuration on Raspberry Pi

Interfaces

Interface Overview

The screenshot shows the 'Interface Overview' page in OpenWRT. It displays a table of network interfaces with their status and actions:

Network	Status	Actions
WAN6 @wan	MAC-Address: 00:00:00:00:00:00 RX: 0.00 B (0 Pkts.) TX: 0.00 B (0 Pkts.)	Connect Stop Edit Delete
WLAN br-WLAN	Uptime: 0h 30m 18s MAC-Address: DE:CD:DE:86:A8:A1 RX: 732.70 KB (8058 Pkts.) TX: 2.06 MB (5146 Pkts.) IPv4: 192.168.1.105/24	Connect Stop Edit Delete
LAN br-lan	Uptime: 0h 0m 0s MAC-Address: B8:27:EB:1A:DC:91 RX: 0.00 B (0 Pkts.) TX: 0.00 B (0 Pkts.)	Connect Stop Edit Delete

A red box highlights the WLAN interface, and a red note below it says 'The two adapters share the same interface'. At the bottom, there is a button 'Add new interface...'.

Figure 5. The interface setting on Raspberry Pi

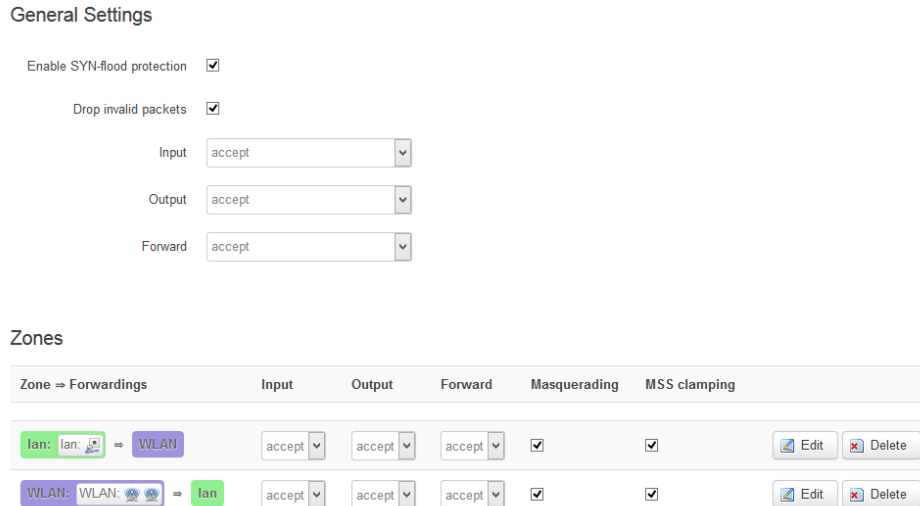


Figure 6. Firewall setting on Raspberry Pi

Reboot everything to make sure the new configuration works fine after every boot. The association is done automatically. Figure 7 and Figure 8 show the status of the wireless network on the base router and the Raspberry Pi, respectively. While there are no wired connection on the Raspberry Pi, the WDS connection provides the access to the base router and the Internet.

The image shows the 'Wireless Nodes' section of a base router interface, divided into 'Clients' and 'WDS Nodes'.

Clients

MAC Address	Interface	Uptime	TX Rate	RX Rate	Signal	Noise	SNR	Signal Quality
14:2D:27:DA:EB:E3	eth1	N/A	N/A	N/A	-65	-93	28	35%
84:A6:C8:CE:58:B5	eth1	N/A	N/A	N/A	-31	-93	62	78%
00:C0:CA:81:F9:15	eth1	N/A	N/A	N/A	-15	-93	78	97%
00:C0:CA:7B:B3:8D	eth1	N/A	N/A	N/A	-14	-93	79	99%

WDS Nodes

MAC Address	Interface	Description	Signal	Noise	SNR	Signal Quality
00:C0:CA:81:F9:15	eth1		-15	-93	78	97%
00:C0:CA:7B:B3:8D	eth1		-14	-93	79	99%
80:1F:02:D3:8E:84	eth1		0	-93	93	0%
74:DA:38:06:65:BD	eth1		0	-93	93	0%

Figure 7. Base router WDS connection

Associated Stations

SSID	MAC-Address	IPv4-Address	Signal	Noise	RX Rate	TX Rate
Jasmine	14:2D:27:DA:EB:E3	?	-66 dBm	0 dBm	1.0 Mbit/s, MCS 0, 20MHz	1.0 Mbit/s, MCS 0, 20MHz
Jasmine	00:40:77:BB:55:12	?	-6 dBm	0 dBm	24.0 Mbit/s, MCS 0, 20MHz	18.0 Mbit/s, MCS 0, 20MHz

WDS access point, i.e. the WLAN port of the base router

Figure 8. OpenWRT WDS connection

4.3 The verification of the seamless handover function

A wireless networks consisting of two Raspberry Pi routers and one base router is set up. This network is depicted in Figure 9. When the whole network is set up, any wireless client should see one SSID, Jasmine in this project, with three different access points as shown in Figure 10.

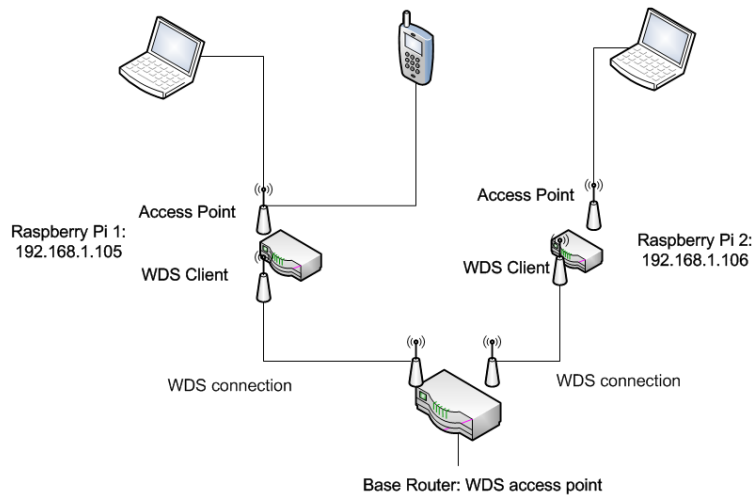


Figure 9. Verification of the design

SSID	SIGNAL	CHANNEL	SECURITY	MAC ADDRESS
★ Jasmine	-32	6	Open	00:40:77:BB:55:12
Jasmine	-29	6	Open	00:C0:CA:7A:BB:5A
Jasmine	-25	6	Open	00:C0:CA:7B:BA:05
Bryan316	-87	1	Open	C8:D7:19:F2:CD:43
wustl-2.0	-73	11	Open	F4:4E:05:7D:C2:50
wustl-guest-2.0	-73	11	Open	F4:4E:05:7D:C2:51

Figure 10. Three access points for the same network

The verification is done using the following 3-step process:

1. The test smartphone is connected to Raspberry Pi 1 (AP MAC: 05) as shown in Figure 11 and started a VoIP session (Zoiper).

2. Manually switch off the access point adapter on Raspberry Pi 1 (AP MAC: 05) as shown in Figure 12. The smartphone automatically reconnected to Raspberry Pi 2 (AP MAC: 5A) as shown in Figure 13.

3. The result is the VoIP session continues even though there is a one-second silent window during the handover because some packets are lost in the handover process.

Wireless Overview

Generic MAC80211 802.11bgn (radio0)
Channel: 6 (2.437 GHz) | Bitrate: 36.6 Mbit/s
SSID: Jasmine | Mode: Master
68% BSSID: 00:C0:CA:7B:BA:05 | Encryption: None

Raspberry Pi 1

Generic MAC80211 802.11bgn (radio1)
Channel: 6 (2.437 GHz) | Bitrate: 18 Mbit/s
SSID: Jasmine | Mode: Client
4% BSSID: 00:40:77:BB:55:12 | Encryption: None

SSID	MAC-Address	IPv4-Address	Signal	Noise	RX Rate	TX Rate
Jasmine	14:2D:27:DA:EB:E3	?	-70 dBm	0 dBm	2.0 Mbit/s, MCS 0, 20MHz	1.0 Mbit/s, MCS 0, 20MHz
Jasmine	C0:65:99:97:37:D7	My phone	-54 dBm	0 dBm	1.0 Mbit/s, MCS 0, 20MHz	72.2 Mbit/s, MCS 7, 20MHz
Jasmine	00:40:77:BB:55:12	?	3 dBm	0 dBm	18.0 Mbit/s, MCS 0, 20MHz	18.0 Mbit/s, MCS 0, 20MHz

Figure 11. My phone connected to Raspberry Pi 1

Wireless Overview

Generic MAC80211 802.11bgn (radio0)
Channel: 6 (2.437 GHz) | Bitrate: 18 Mbit/s
SSID: Jasmine | Mode: Unknown
0% Wireless is disabled or not associated

Generic MAC80211 802.11bgn (radio1)
Channel: 6 (2.437 GHz) | Bitrate: 18 Mbit/s
SSID: Jasmine | Mode: Client
100% BSSID: 00:40:77:BB:55:12 | Encryption: None

SSID	MAC-Address	IPv4-Address	Signal	Noise	RX Rate	TX Rate
Jasmine	00:40:77:BB:55:12	?	-4 dBm	0 dBm	24.0 Mbit/s, MCS 0, 20MHz	18.0 Mbit/s, MCS 0, 20MHz

Figure 12. The access point on Raspberry Pi 1 is manually turned off

Generic MAC80211 802.11bgn (radio0)
Channel: 6 (2.437 GHz) | Bitrate: 72.2 Mbit/s
SSID: Jasmine | Mode: Master
88% BSSID: 00:C0:CA:7A:BB:5A | Encryption: None

Raspberry Pi 2

Generic MAC80211 802.11bgn (radio1)
Channel: 6 (2.437 GHz) | Bitrate: 18 Mbit/s
SSID: Jasmine | Mode: Client
100% BSSID: 00:40:77:BB:55:12 | Encryption: None

SSID	MAC-Address	IPv4-Address	Signal	Noise	RX Rate	TX Rate
Jasmine	C0:65:99:97:37:D7	192.168.1.102	-48 dBm	0 dBm	1.0 Mbit/s, MCS 0, 20MHz	72.2 Mbit/s, MCS 7, 20MHz
Jasmine	00:40:77:BB:55:12	?	-18 dBm	0 dBm	54.0 Mbit/s, MCS 0, 20MHz	18.0 Mbit/s, MCS 0, 20MHz

Figure 13. The smartphone is reconnected to Raspberry Pi 2

This result verifies that this design avoids VoIP phone call drop during the handover process.

This verification method is a simulation of the real world scenario:

A new UAV flies to the position to replace the old one. The access point on the old one is manually switched off after the new UAV has settled down. All the clients will be reconnected to the new access point automatically without any possible communication service interruption. Then we can retrieve the old UAV for battery recharge.

Another possible scenario is that the client moves from one UAV to the other one. The seamless handover is still valid. What will happen is the signal strength of the old one keeps dropping down until the connection is not stable and the reconnect request is made. The wireless client will be connected to the UAV automatically as it has a better signal strength for the same wireless network.

5. Field Test of the Design

Some field tests were done to check if this design is doable in the UAV application. Coverage and signal strength test is to see whether it is suitable for long range connection in open space. Network speed test is to see what is the throughput performance of the design. Finally, switch-over time is to verify the seamless handover function with numbers.

5.1 Coverage and Signal Strength

Field tests on the signal strength of several possible options in open space environment to simulate the situation on UAV were conducted. The experiment setting is illustrated in Figure 14. The WiFi access point is set at an open window on the second floor of a multi-apartment building facing a road and a grass land. All experiments are done with no vehicle passing by which may interfere with the WiFi signal.

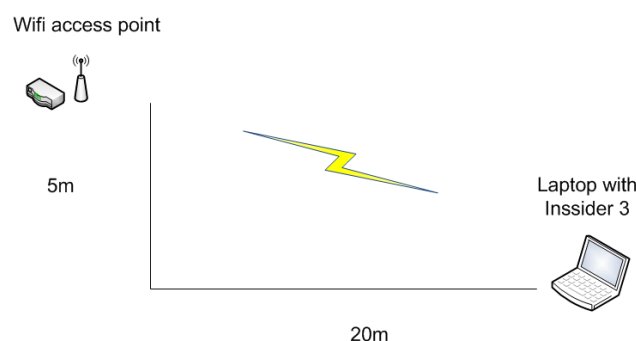


Figure 14. The setting of the field test on signal strength

Part of the results are listed in Table 1. These are some important combinations which may be the used in the actual system. From the results we see that in long

distance in open space, the signal strength is pretty strong and stable. Also the result of Bryan sky, which was the commercial router used on the UAV to prove the conception, were put in the table for comparison purpose. The particular router had very good signal performance in the sky.

Table 1. The field test results on signal strength

Router	Antenna Size	Signal Strength (dBm): Average of 15 test results
RPi + Alfa	Regular	-58.5
RPi + Alfa	Long 1	-52.8
Linksys	Long 2	-60.4
Linksys	Directional	-52.7
Bryan Sky	-----	-49

RPi + Alfa: Raspberry Pi + OpenWRT+ Alfa AWUS036NHR v2

Linksys: Linksys WRT54G v5 router

Regular: The default antenna with the Alfa adapters

Long 1: The large size antenna for Alfa adapters, expected to be 5 dBm better

Long 2: The large size antenna for Linksys router.

Directional: The directional antenna used on Linksys router, to increase signal strength in one direction.

Bryan Sky: The commercial router for comparison.

The result shows that Raspberry Pi with better antenna can achieve very good signal strength in the environment setting of the project. On the other hand, the Linksys router which is used as the base router had similar coverage with directional antenna. This is not very satisfying as it is supposed to be more powerful and be the central station of the whole system.

The sensitivity of these routers is also verified. In general case, when the signal strength is lower than -80 dBm, the quality of the connection becomes unacceptable. This is much lower than the results in Table 1.

One of the students in the course “Computer Systems Analysis” has done a deep analysis on the data we achieved as his project. You can retrieve the report on the course website when it is finished [Jain15].

5.2 Network speed

The half-duplex nature of WDS mode limits the overall speed it can achieve. In theory it will cut the speed to half. Table 2 is the result using speed testing APP (SpeedTest) on a test laptop. For comparison, five more tests were conducted after the WDS connection was replaced with the wired connection. The result is listed in Table 3. The comparison between the two tables shows the effect of the half-duplex working mode. The upload and download speed was reduced by 52% and 74% respectively. Although the speed is not ideal, it is sufficient for most VoIP application.

Replacing Raspberry Pi with a more powerful platform, such as Raspberry Pi II which was just released in March 2015, is a viable option to improve the speed performance. Regarding WDS connection, there is not too much we can do. It was designed years ago and not suitable for high speed communication.

Table 2. Speed Test of WDS network connection (5 tests)

Test Number	1	2	3	4	5
Ping (ms)	5	5	5	5	5
Download (Mbps)	0.62	0.68	0.69	0.66	0.71
Upload (Mbps)	1.79	2.12	2.10	1.67	1.57

Table 3. Speed Test of wired connection instead of the WDS connection (5 tests)

Test Number	1	2	3	4	5
Ping (ms)	5	4	5	5	4
Download (Mbps)	2.62	2.46	2.58	2.61	2.54
Upload (Mbps)	3.78	3.62	3.97	4.01	3.76

5.3 Switch over time

In WiFi handover, switch over time is an important factor. It shows that the reconnection is quick enough to avoid the Transmission Control Protocol (TCP) session termination as the TCP session is able to be kept alive in the handover process. Five tests are done. Packets on the WLAN port on a laptop during handover were checked via WireShark to find the gap between two WiFi connection sessions. The result is listed in table 4.

Table 4. Switch over time test results.

Test Number	1	2	3	4	5
Switch over time (s)	0.28	0.21	0.17	0.25	0.27

6. Future Work

All the work have been done are preliminary. More field tests including the one on UAVs are required. The different environment on UAV will cause some unexpected problem. The vibration and the power supply are examples of the unknown issues. Speed limit and the stability issue of WDS mode need to be addressed in the future. It could cause major service quality degradation.

On the other hand, new products provide more options and possible solutions to the issues we have met. For instance, the newly released Raspberry Pi II is much more powerful than the one we are using in this project. The extra power could increase the coverage, stability and speed performance.

7. Summary

In this project, the seamless handover function between WiFi stations mounted on UAVs was realized. Although there are some issues such as stability and speed limit, the whole design provides sufficient signal strength to sustain a stable VoIP connection and keep it alive during handover. Field tests, which simulate the real world application environment, have been done to verify this design.

The necessary steps to implement OpenWRT and WDS mode on Raspberry Pi are illustrated in detail. All necessary firmware and downloads are available online. All the WiFi related equipment used in this project are common home level commercial products. Once the configuration has been done correctly, the network will be automatically on as long as all necessary equipment are powered properly.

8. Acronyms

DHCP	Dynamic Host Configuration Protocol
TCP	The Transmission Control Protocol
UAV	Unmanned air vehicle
VoIP	Voice over IP
WDS	Wireless Distribution System

9. References

[RPi1] "OpenWRT on Raspberry Pi", http://wiki.openwrt.org/toh/raspberry_pi

[RPi2] "The implementation steps of OpenWRT on Raspberry Pi",
<http://fullcircle.so/install-openwrt-raspberry-pi/>

[wiki1] "Wikipedia of WDS mode",
http://en.wikipedia.org/wiki/Wireless_distribution_system

[WDS1] "WDS mode on OpenWRT", <http://wiki.openwrt.org/doc/howto/clientmode>

[LUCI] "Configuration interface of OpenWRT",
<http://wiki.openwrt.org/doc/techref/luci>

[OPKG] "Software installation package on OpenWRT",
<http://wiki.openwrt.org/doc/techref/opkg>

[Jain15] "Course website of Computer Systems Analysis",
<http://www.cse.wustl.edu/~jain/cse567-15/index.html>