Alexis Allen Studio

Scripts for setting up Wi-Fi routers for 3D room scanning.

---

1. Flash OpenWRT or DD-WRT on the Routers

Step 1.1: Check Compatibility

Run this command on your terminal to identify your router's model:
bash
Copy code
```
sudo nmap -sP 192.168.1.0/24
```

- This scans your local network for devices. Identify your router's IP and MAC address.

Step 1.2: Download Firmware

- Visit the OpenWRT Hardware Table or DD-WRT Router Database.
- Download the firmware file compatible with your router model. Make sure the file format is `.bin`.

Step 1.3: Flash Firmware

Using the Router’s Web Interface:
Log in to your router’s admin panel:
bash
Copy code
```
firefox http://192.168.1.1
```

1.
2. Navigate to the **Firmware Upgrade** section.
3. Upload the `.bin` file and click **Upgrade**.

Using a TFTP Server:
Install a TFTP client:
bash
Copy code
```
sudo apt install tftp
```

1.

Transfer the firmware file to the router:
bash

Copy code
```
tftp 192.168.1.1
put firmware.bin
```

2.
3. Wait for the router to reboot.

---

2. Configure the Routers

Step 2.1: Set Router to Access Point Mode (Transmitter)

Access the router's admin interface:
bash
Copy code
```
firefox http://192.168.1.1
```

1.
2. For OpenWRT:
    - Go to **Network > Wireless**.

Set the mode to **Access Point**:
bash
Copy code
```
uci set wireless.@wifi-device[0].mode='ap'
uci commit wireless
```

    -

Apply settings:
bash
Copy code
```
wifi reload
```

    -
3. For DD-WRT:
    - Navigate to **Wireless > Basic Settings**.
    - Set **Wireless Mode** to "AP."
    - Save and reboot.

Step 2.2: Set Router to Client Mode (Receiver)

1. For OpenWRT:

- Go to **Network > Wireless**.

Set the mode to **Client**:
bash
Copy code
```
uci set wireless.@wifi-device[0].mode='sta'
uci set wireless.@wifi-iface[0].ssid='YOUR_AP_SSID'
uci set wireless.@wifi-iface[0].key='YOUR_AP_PASSWORD'
uci set wireless.@wifi-iface[0].encryption='psk2'
uci commit wireless
```

-

Apply settings:
bash
Copy code
```
wifi reload
```

-

2. For DD-WRT:
   - Navigate to **Wireless > Basic Settings**.
   - Set **Wireless Mode** to "Client."
   - Enter the SSID and password of the transmitter AP.
   - Save and reboot.

---

3. Enable SSH or Telnet

Step 3.1: Enable SSH

OpenWRT:
Enable SSH from the web interface:
bash
Copy code
```
uci set dropbear.@dropbear[0].enable='1'
uci commit dropbear
/etc/init.d/dropbear restart
```

1.

DD-WRT:

1. Navigate to **Services > Services**.
2. Enable **SSHd** and set the port (default is 22).
3. Save and reboot.

Step 3.2: Enable Telnet

OpenWRT:
Enable Telnet manually:
bash
Copy code
```
/etc/init.d/telnet enable
/etc/init.d/telnet start
```

-

DD-WRT:

- Navigate to **Services > Services**.
- Enable Telnet and save changes.

---

4. Collect CSI or Signal Metrics

Step 4.1: Install CSI Tools

Clone and compile the Intel CSI Tool:
bash
Copy code
```
git clone
https://github.com/dhalperi/linux-80211n-csitool-supplementary.git
cd linux-80211n-csitool-supplementary/netlink
make
```

1.

Enable CSI collection:
bash
Copy code
```
modprobe ath9k
```

2.

Run the CSI tool:
bash
Copy code
```
./log_to_file wlan0 output_file.dat
```

3.

Step 4.2: Capture Packets with Wireshark or TShark

Install Wireshark and TShark:
bash
Copy code
```
sudo apt install wireshark tshark
```

1.

Start a capture on the Ethernet interface:
bash
Copy code
```
tshark -i eth0 -w capture.pcap
```

2.

Step 4.3: Process Captured Data

Extract CSI:
python
Copy code
```
import numpy as np

# Load CSI data
csi_data = np.load("output_file.dat")

# Analyze phase and amplitude
amplitude = np.abs(csi_data)
phase = np.angle(csi_data)
print("Amplitude:", amplitude)
print("Phase:", phase)
```

1.

---

5. Process and Visualize Data

Step 5.1: Process Data for 3D Reconstruction

Use Fourier transforms to analyze multipath signals:
python
Copy code

```
from scipy.fft import fft

# Perform FFT
freq_domain = fft(amplitude)
print("Frequency Domain Data:", freq_domain)
```

1.

Step 5.2: Create 3D Visualizations

Use `matplotlib` or `pyvista`:
python
Copy code

```
import pyvista as pv
import numpy as np

# Example 3D grid
grid_x, grid_y, grid_z = np.meshgrid(
    np.linspace(-5, 5, 50),
    np.linspace(-5, 5, 50),
    np.linspace(-5, 5, 50)
)
# Map amplitude to grid points
scalar_field = amplitude.reshape(50, 50, 50)

# Create a 3D plot
grid = pv.StructuredGrid(grid_x, grid_y, grid_z)
grid["Amplitude"] = scalar_field.flatten()
```

1. `grid.plot(volume=True)`