Applied Industrial Internet Of Things

Project Statement-1

Wifi Range Extender Using Raspberry Pi

Mid-Course Project

Batch 49(4)

Aim:-

Develop a Wi-Fi Range Extender Using Raspberry Pi to improve wireless coverage in a specified area, overcoming dead zones and enhancing connectivity. The solution should be cost-effective, easy to deploy, and compatible with standard Wi-Fi protocols.

Problem statement:-

Materials Needed:

Raspberry Pi (any model with Wi-Fi capabilities)

MicroSD card (8GB or larger)

Power supply for Raspberry Pi

Ethernet cable (for initial setup)

Wi-Fi dongle (optional, if your Raspberry Pi doesn't have built-in Wi-Fi)

External antenna (optional, for better range)

Instructions:

1.Prepare Raspberry Pi:

Download and install the latest version of Raspberry Pi OS on the MicroSD card.

Insert the MicroSD card into the Raspberry Pi and connect it to power.

If your Raspberry Pi doesn't have built-in Wi-Fi, connect the Wi-Fi dongle.

2.Initial Configuration:

Connect the Raspberry Pi to your router using an Ethernet cable.

Power on the Raspberry Pi.

SSH into the Raspberry Pi or access it via a monitor and keyboard.

Update the system: sudo apt update && sudo apt upgrade.

3.Install Hostapd and DNSMasq:

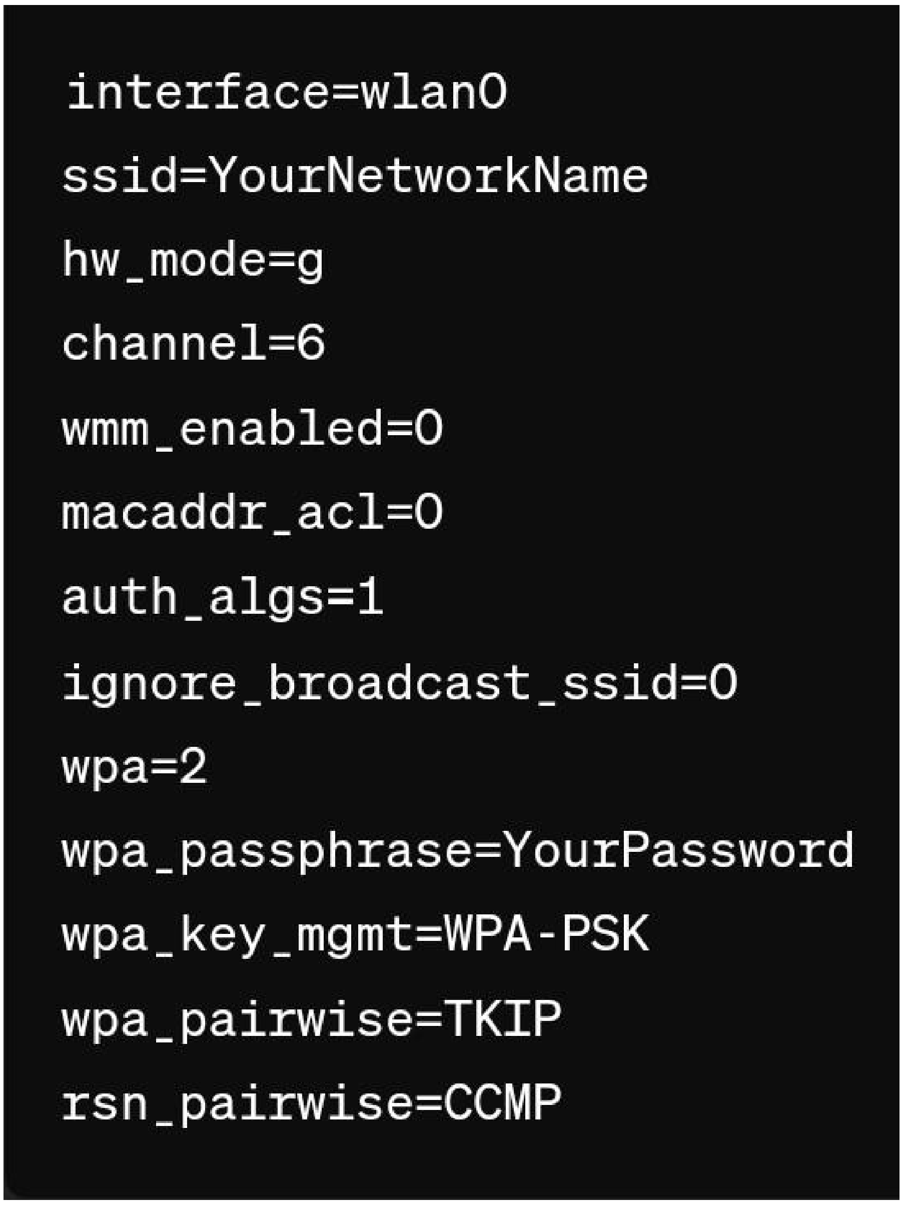
Hostapd allows you to create a Wi-Fi hotspot, while DNSMasq provides DNS and DHCP services.

Install Hostapd and DNSM asq: sudo apt install hostapd dnsmasq.

4.Configure Hostapd:

Edit the Hostapd configuration file: sudo nano /etc/hostapd/hostapd.conf.

Configure the Wi-Fi network settings:



```
interface=wlan0
ssid=YourNetworkName
hw_mode=g
channel=6
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=YourPassword
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

5.Configure DNSMasq:

Edit the DNSMasq configuration file: sudo nano /etc/dnsmasq.conf.Add the following lines at the end of the file:

```
interface=wlan0
dhcp-range=192.168.4.2,192.168.4.20,255.255.255.0,24h
```

6.Enable IP Forwarding:

Edit the sysctl.conf file: sudo nano /etc/sysctl.conf.

Uncomment the line #net.ipv4.ip_forward=1 by removing the #.

Save and exit the file.

7. Routing and NAT:

Enable routing: sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE.

Save the iptables rule: sudo sh -c "iptables-save > /etc/iptables.ipv4.nat".

Edit the rc.local file: sudo nano /etc/rc.local.

Add the following line above exit 0: iptables-restore < /etc/iptables.ipv4.nat.

Save and exit the file.

8. Reboot Raspberry Pi:

Reboot the Raspberry Pi: sudo reboot.

9. Final Steps:

Once the Raspberry Pi restarts, it should broadcast the Wi-Fi network you configured.

Connect your devices to this network, and you should have extended Wi-Fi coverage in the area.

10. Optional:

You can further enhance the range by using an external antenna and positioning it for optimal coverage.

By following these steps, you can create a cost-effective Wi-Fi Range Extender using a Raspberry Pi, improving wireless coverage in your specified area, overcoming dead zones, and enhancing connectivity.

Scope of the solution:

The scope of the Wi-Fi range extender solution using Raspberry Pi includes:

1.Hardware Selection: Choose appropriate Raspberry Pi model (e.g., Raspberry Pi 4), Wi-Fi dongle or USB Wi-Fi adapter, microSD card, power supply, and optional accessories like an enclosure.

2.Operating System Installation: Install Raspberry Pi OS Lite or another lightweight Linux distribution suitable for the Raspberry Pi.

3.Initial Configuration: Set up the Raspberry Pi, including network configuration to connect to the existing Wi-Fi network.

4.Software Installation: Install necessary software packages such as hostapd and dnsmasq to enable the Raspberry Pi to act as a Wi-Fi access point.

5.Configuration of Access Point: Configure hostapd and dnsmasq to set up the SSID, passphrase, Wi-Fi channel, and DHCP settings for the new access point.

6.Testing: Ensure that devices can connect to the new Wi-Fi network created by the Raspberry Pi and verify internet connectivity.

Required components to develops solution:

Hardware:

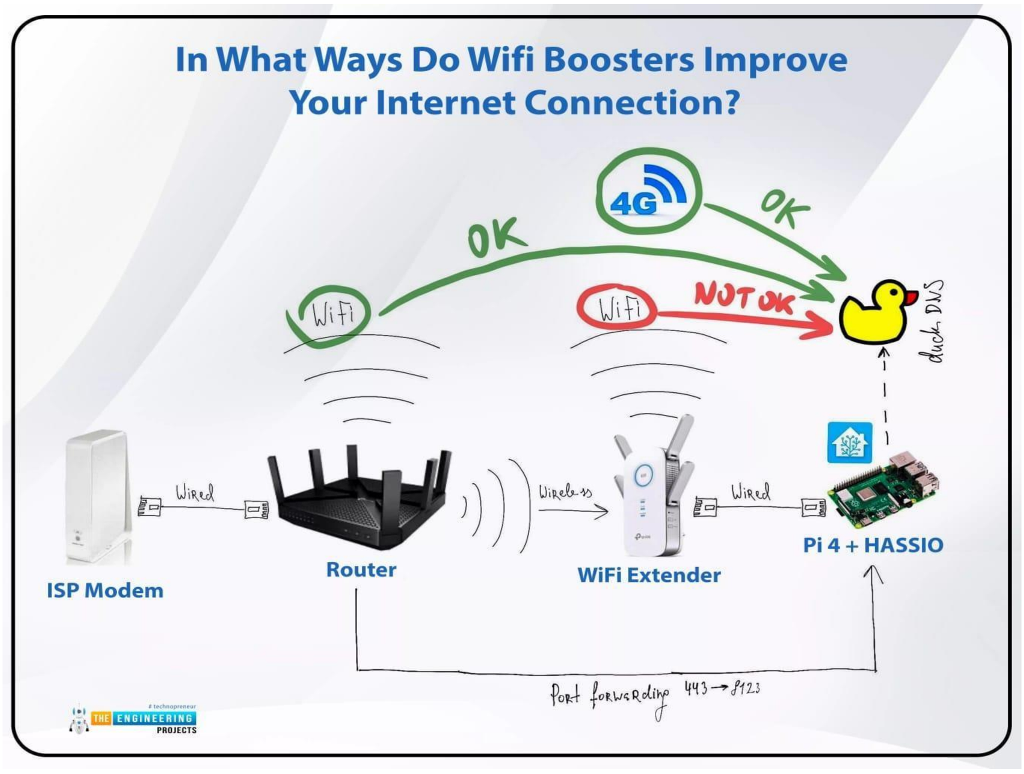
1. **Raspberry Pi**: Raspberry Pi 4 Model B (or newer) - This will serve as the main computing unit.
2. **Wi-Fi Dongle or USB Wi-Fi Adapter**: TP-Link AC600 USB Wi-Fi Adapter - Provides wireless connectivity for the Raspberry Pi.
3. **MicroSD Card**: SanDisk Ultra 32GB micro SDHC - Used to store the operating system and software.
4. **Power Supply**: Cana Kit Raspberry Pi 4 Power Supply - Provides power to the Raspberry Pi.
5. **Ethernet Cable (Optional)**: Amazon Basics RJ45 Cat-6 Ethernet Patch Cable - Used for initial setup and configuration.
6. **Enclosure (Optional)**: Official Raspberry Pi 4 Case - Provides protection for the Raspberry Pi.

Software:

1. **Raspberry Pi OS Lite**: Lightweight operating system optimized for Raspberry Pi. It can be downloaded from the official Raspberry Pi website.
2. **Hostapd**: Software package that allows the Raspberry Pi to act as a Wi-Fi access point. It can be installed using the package manager (sudo apt install hostapd).
3. **dnsmasq:** A lightweight DNS and DHCP server. It is used to assign IP addresses to devices connecting to the Wi-Fi network

created by the Raspberry Pi. It can be installed using the package manager (sudo apt install dnsmasq).

Simulated circuit (Tinkercad/Fritzing):



Gerber file:

Hardware Setup:

1. Raspberry Pi: Choose a Raspberry Pi model with built-in Wi-Fi capabilities (e.g., Raspberry Pi 3 or newer).

2. Wi-Fi Dongle (optional): If your Raspberry Pi model doesn't have built-in Wi-Fi or you want to use a more powerful Wi-Fi adapter, you can use a USB Wi-Fi dongle.

3. Power Supply: Make sure you have a stable power supply for the Raspberry Pi.

4. enclosure (optional): Consider housing your Raspberry Pi in an enclosure for protection.

Software Setup:

Install Raspberry Pi OS (formerly Raspbian) on the Raspberry Pi.

Connect the Raspberry Pi to your network via Ethernet cable or Wi-Fi.

Update the OS and install necessary packages:

sql

Copy code

```
sudo apt update

sudo apt upgrade

sudo apt install hostapd dnsmasq
```

Configure Wi-Fi Access Point (AP):

Configure the Raspberry Pi to act as a Wi-Fi AP. You can use hostapd to create the AP configuration.

Configure DHCP server using dnsmasq to assign IP addresses to devices connecting to the AP.

Bridge Connections:

Bridge the Wi-Fi connection from the Raspberry Pi to the existing Wi-Fi network to extend its range. You can use iptables or bridge-utils for this purpose.

Test and Deployment: Test the range extender in different locations to ensure coverage improvement.

Consider deploying multiple range extenders strategically to cover dead zones effectively.

Monitoring and Maintenance:

Monitor the performance of the range extender periodically to ensure it' s functioning properly.

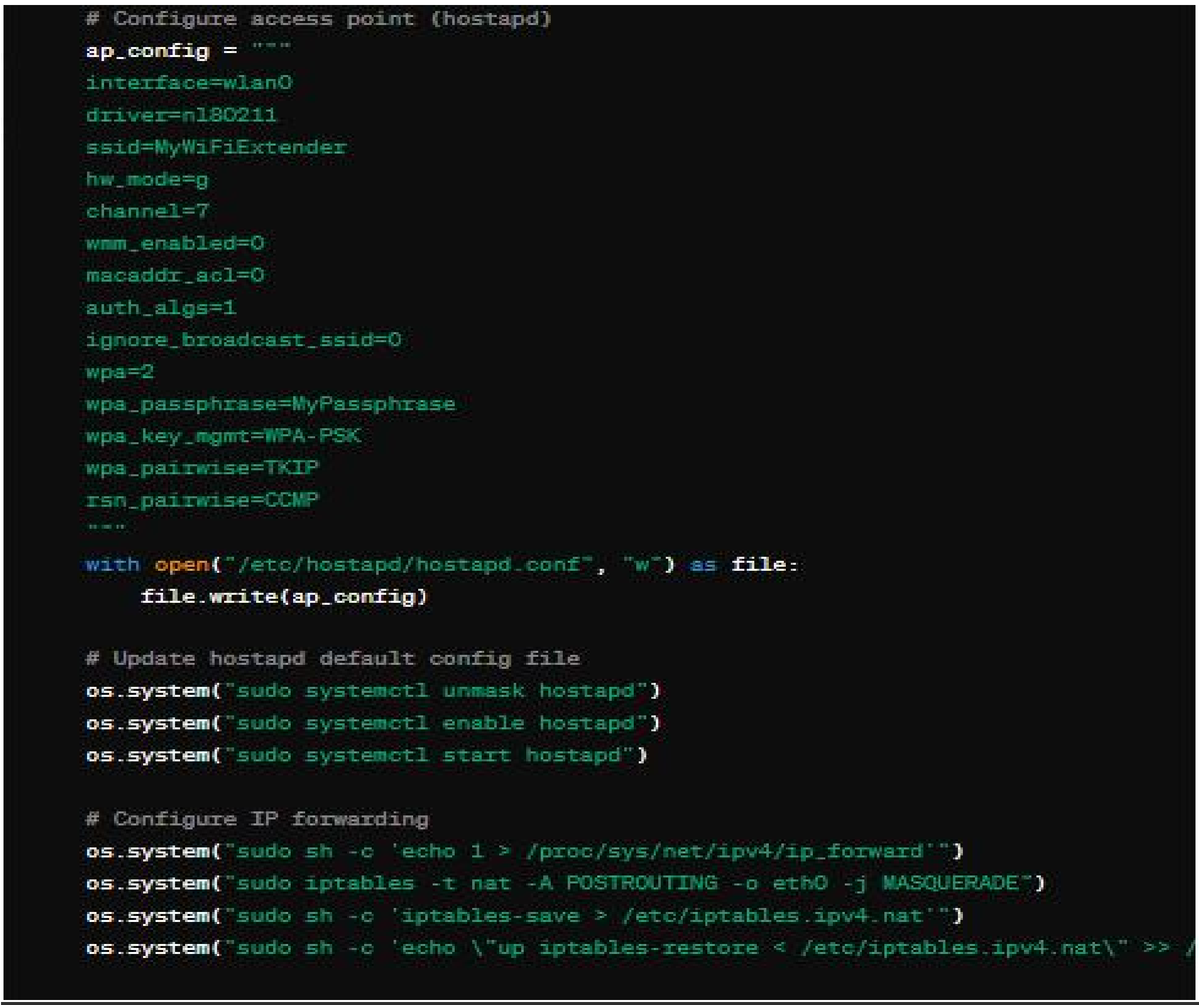
Update the software and firmware of the Raspberry Pi regularly to patch security vulnerabilities and improve stability.

Remember to document your setup and configurations for future reference. Additionally, ensure compliance with local regulations and network policies while deploying Wi-Fi range extenders.

Code for the solution:

1. Fabricate
2. Design
3. Construct
4. Generate
5. Formulate

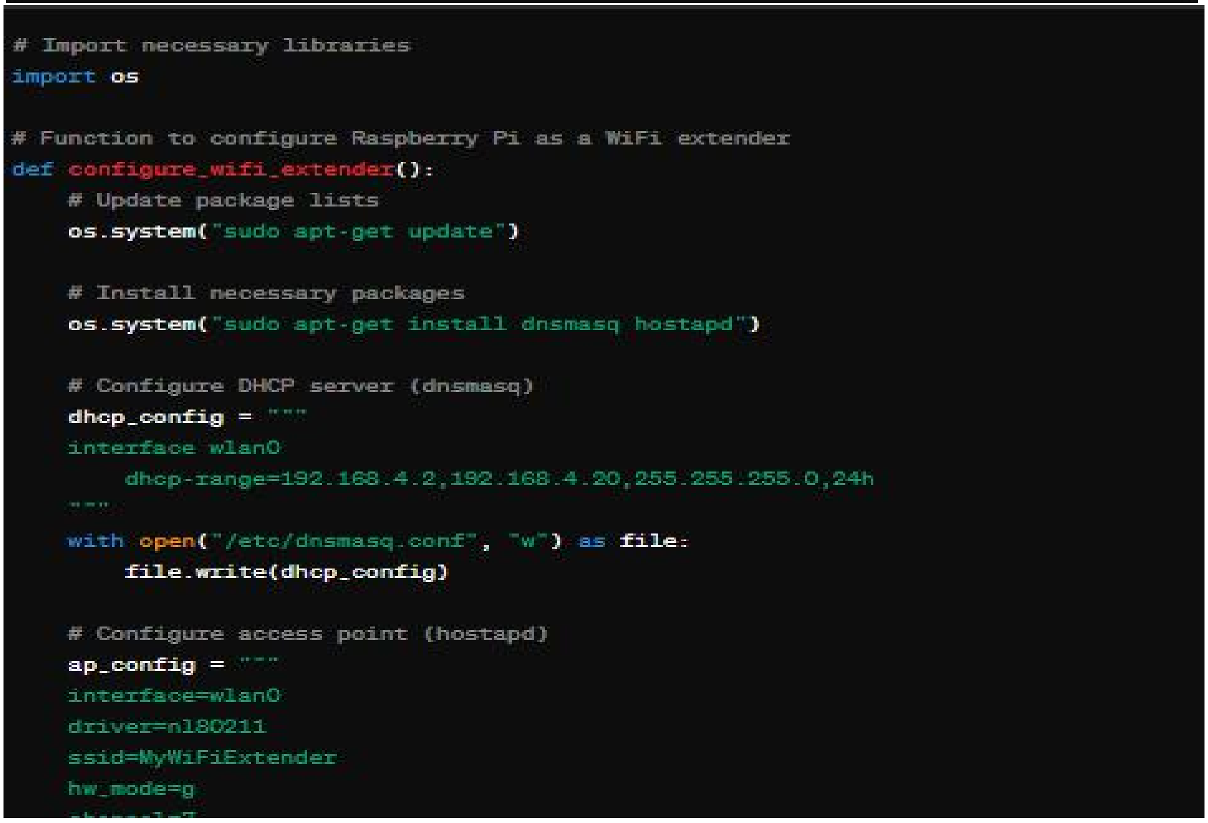
Example: "Create a Wi-Fi Range Extender Using Raspberry Pi to enhance wireless coverage in a designated space, eliminating dead zones and amplifying connectivity. The solution must be economical, simple to implement, and compliant with standard Wi-Fi protocols. Write the code for the solution."



```
    # Configure access point (hostapd)
    ap_config = """
    interface=wlan0
    driver=nl80211
    ssid=MyWiFiExtender
    hw_mode=g
    channel=7
    wmm_enabled=0
    macaddr_acl=0
    auth_algs=1
    ignore_broadcast_ssid=0
    wpa=2
    wpa_passphrase=MyPassphrase
    wpa_key_mgmt=WPA-PSK
    wpa_pairwise=TKIP
    rsn_pairwise=CCMP
    """
    with open("/etc/hostapd/hostapd.conf", "w") as file:
        file.write(ap_config)

    # Update hostapd default config file
    os.system("sudo systemctl unmask hostapd")
    os.system("sudo systemctl enable hostapd")
    os.system("sudo systemctl start hostapd")

    # Configure IP forwarding
    os.system("sudo sh -c 'echo 1 > /proc/sys/net/ipv4/ip_forward'")
    os.system("sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE")
    os.system("sudo sh -c 'iptables-save > /etc/iptables.ipv4.nat'")
    os.system("sudo sh -c 'echo \"up iptables-restore < /etc/iptables.ipv4.nat\" >> /
```



```
# Import necessary libraries
import os

# Function to configure Raspberry Pi as a WiFi extender
def configure_wifi_extender():
    # Update package lists
    os.system("sudo apt-get update")

    # Install necessary packages
    os.system("sudo apt-get install dnsmasq hostapd")

    # Configure DHCP server (dnsmasq)
    dhcp_config = """
    interface wlan0
        dhcp-range=192.168.4.2,192.168.4.20,255.255.255.0,24h
    """
    with open("/etc/dnsmasq.conf", "w") as file:
        file.write(dhcp_config)

    # Configure access point (hostapd)
    ap_config = """
    interface=wlan0
    driver=nl80211
    ssid=MyWiFiExtender
    hw_mode=g
```

Conclusion:

In conclusion, developing a Wi-Fi range extender using a Raspberry Pi offers a cost-effective, easy-to-deploy solution for improving wireless coverage in specified areas, overcoming dead zones, and enhancing connectivity. By leveraging standard Wi-Fi protocols and the versatility of the Raspberry Pi platform, users can extend the range of their existing network without the need for expensive commercial equipment.

This solution provides flexibility in deployment, allowing users to strategically place range extenders to maximize coverage where needed most. Additionally, the Raspberry Pi's accessibility and robust community support make it an ideal choice for DIY enthusiasts and small businesses looking to enhance their Wi-Fi networks without breaking the bank.

Team members :

1) Suddula Manjula
2) Vadla Bhavani
3) Y.Lavanya
4) Seege Mounika