

Analysis and Implementation of Functionality for Modular Router

Major Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology

in

Electronics & Communication Engineering

(Embedded Systems)

By

Bhavika Joshi

(15MECE03)



Electronics & Communication Engineering Department

Institute of Technology

Nirma University

Ahmedabad-382 481

May 2017

Analysis and Implementation of Functionality for Modular Router

Major Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology

in

Electronics & Communication Engineering

(Embedded Systems)

By

Bhavika Joshi

(15MECE03)



Under the guidance of

External Project Guide:

Chaitanya Naredla

Director Engineering

Smartron India Pvt. Ltd.,

Hyderabad.

Internal Project Guide:

Prof. Vijay Savani

Assistant Professor, EC Department,

Institute of Technology,

Nirma University, Ahmedabad.

Electronics & Communication Engineering Department

Institute of Technology

Nirma University

Ahmedabad-382 481

Declaration

This is to certify that

- a. The thesis comprises my original work towards the degree of Master of Technology in Embedded Systems at Nirma University and has not been submitted elsewhere for a degree.
- b. Due acknowledgment has been made in the text to all other material used.

- **Bhavika Joshi**

15MECE03

Disclaimer

”The content of this paper does not represent the technology, opinions, beliefs, or positions of Smartron India Pvt. Ltd., its employees, vendors, customers, or associates.”



Certificate

This is to certify that the Major Project entitled “**Analysis and Implementation of Functionality for Modular Router**” submitted by **Bhavika Joshi (15MECE03)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in Embedded Systems, Nirma University, Ahmedabad is the record of work carried out by her under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of our knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Date:

Place: Ahmedabad

Prof. Vijay Savani

Internal Guide

Program Coordinator

Dr. D.K.Kothari

Section Head, EC

Dr. Alka Mahajan

Director, IT



Certificate

This is to certify that the Major Project entitled “**Analysis and Implementation of Functionality for Modular Router**” submitted by **Bhavika Joshi(15MECE03)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in Embedded Systems, Nirma University, Ahmedabad is the record of work carried out by her under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination.

Chaitanya Naredla
Director Engineering
Smartron India Pvt. Ltd.
Hyderabad

Acknowledgements

I would like to express my gratitude and sincere thanks to **Dr. N.P Gajjar**, PG Coordinator of M.Tech Embedded Systems program for allowing me to undertake this thesis work and for his guidelines during the review process.

I take this opportunity to express my profound gratitude and deep regards to **Prof. Vijay Savani**, guide of my major project for his exemplary guidance, monitoring and constant encouragement throughout the course of this thesis. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark. I would take this opportunity to express a deep sense of gratitude to **Chaitanya Naredla, Vineeth Paruchuri**, Engineering Manager, Smartron India Pvt. Ltd. for his cordial support, constant supervision as well as for providing valuable information regarding the project and guidance, which helped me in completing this task through various stages. I would also thank to **Shyam Gopaldasamy, Shivram Rajan Velmurgan**, my Project Mentor for always helping, giving me good suggestions, solving my doubts and guide me to complete my project in better way.

I am obliged to **Subhash Kumar, Karthikeyan Prakash** team member of t-home team, Smartron India Pvt. Ltd. for the valuable information provided by his in respective fields. I am grateful for his cooperation during the period of my assignment.

Lastly, I thank almighty, my parents, brother and friends for their constant encouragement without which this assignment would not be possible.

- **Bhavika Joshi**

15MECE03

Abstract

Significant growth in IoT and Home Automation in developed and developing nations opens market for huge potential products. HUB acts as a main gateway for Home Automation products. Modular router bring additional functionality to use the Home Automation HUB as router by eliminating the traditional non-modular router feature.

To integrate all appliances in smart home such as music centers, set top boxes and make their functionality better, wifi router presence is extremely valuable in home automation.

Developed all functionality of router on hub so that we will able to control devices with smartphone or any other device and automate all smart home appliances activity. Developed different wifi router modules functionality such as hostapd, dhcp server, iptables, network configuration, wpa supplicant. For providing service of wifi network connectivity to n thing devices in smarthome, I developed wifi access point in intel nuc board using hostapd, dhcp, iptable, iprouting module and also developed boot time and installation script for wifi access point. Developed application to identify number of devices connected to wifi access point, by using iptable firewall rule ,developed a rule for all illegal devices who are trying to connect to our wifi access point ,restrict their connection. For backend process , Developed API for web user interface of router. Developed backend API related to ifconfig, iwconfig, iwlist, iwscan, wpa_cli, wpa_supplicant, hostapd, dhcp module.

Contents

| | |
|--|----------|
| Declaration | iii |
| Disclaimer | iv |
| Certificate | v |
| Certificate | vi |
| Acknowledgements | vii |
| Abstract | viii |
| List of Tables | xii |
| List of Figures | xiv |
| 1 Introduction | 1 |
| 1.1 Motivation | 1 |
| 1.2 Objective | 1 |
| 1.3 Scope | 1 |
| 1.4 Requirements | 2 |
| 1.5 Gantt Chart | 2 |
| 1.6 Router Introduction | 2 |
| 1.7 Feature to increase speed of wireless router | 5 |
| 1.7.1 Flash memory | 5 |

| | | |
|----------|-------------------------------------|-----------|
| 1.7.2 | RAM | 5 |
| 1.7.3 | Radios | 5 |
| 1.7.4 | CPU | 6 |
| 2 | Literature review | 7 |
| 2.1 | Router | 7 |
| 2.2 | Analysis on Router | 7 |
| 2.3 | Router feature | 12 |
| 2.3.1 | Router QOS | 12 |
| 3 | Tools of Router | 14 |
| 3.0.1 | Dynamic Host Configuration Protocol | 14 |
| 3.0.2 | Domain Name System | 15 |
| 3.0.3 | HostAPD | 17 |
| 3.0.4 | IW | 22 |
| 3.0.5 | wpa_supplicant | 23 |
| 3.0.6 | wpa_cli | 24 |
| 4 | OpenWRT Framework | 26 |
| 4.1 | Introduction | 26 |
| 4.2 | OpenWRT Features | 27 |
| 4.3 | Building an Image | 27 |
| 4.4 | Banana Pi R1 | 29 |
| 5 | IPtables | 30 |
| 5.1 | Block diagram of IPtables firewall | 30 |
| 5.2 | Tables | 31 |
| 5.3 | Targets and Jumps | 32 |
| 5.4 | IPtables Switch Commands | 32 |
| 5.5 | IPtables rule | 34 |

| | |
|---|-----------|
| <i>CONTENTS</i> | xi |
| 6 Analysis of Web user Interface | 38 |
| 6.1 Webmin | 38 |
| 7 Wifi Provider for thing Devices | 40 |
| 7.1 HostAPD | 40 |
| 7.2 Hostapd Configuration | 41 |
| 7.3 Network Interface File | 43 |
| 7.4 DHCP server | 44 |
| 7.5 DHCP Configuration | 45 |
| 7.6 Network Address Translation | 46 |
| 8 thome RESTful API Development Standard | 50 |
| 8.1 Technology | 50 |
| 8.2 Development Environment | 51 |
| 8.3 Documentation | 51 |
| 8.4 Testing | 52 |
| 9 Conclusion | 55 |
| 10 Future Scope | 56 |
| Bibliography | 57 |

List of Tables

- 3.1 Frequency Band [17] 18
- 3.2 802.11 [17] 19

- 5.1 IPtables switch commands [7] 33

List of Figures

| | | |
|-----|---|----|
| 1.1 | Gantt Chart | 2 |
| 2.1 | Almond touch screen display[1] | 9 |
| 2.2 | Cloud service architecture in chime[2] | 10 |
| 2.3 | Torch Application parental control feature[3] | 11 |
| 2.4 | Netgear router QOS[15] | 13 |
| 3.1 | DHCP Flow | 15 |
| 3.2 | Supported interface mode | 17 |
| 3.3 | Network Driver Details[2] | 20 |
| 3.4 | iw_list | 22 |
| 3.5 | iw_list | 25 |
| 4.1 | Banana Pi R1 image [14] | 29 |
| 5.1 | Block diagram [10] | 30 |
| 5.2 | iptables rule | 35 |
| 5.3 | iptables rule | 36 |
| 5.4 | iptables rule | 37 |
| 6.1 | Webmin configuration [13] | 39 |
| 7.1 | Mobile connected with enable Access Point | 41 |
| 7.2 | Hostapd configuration file | 42 |

| | | |
|-----|---|----|
| 7.3 | Network Interface file | 43 |
| 7.4 | Mobile device connected to access point | 44 |
| 7.5 | DHCP configuration file | 45 |
| 7.6 | Mobile device connected to access point | 46 |
| 7.7 | Mobile device connected to access point | 47 |
| 7.8 | Mobile device connected to access point | 48 |
| 7.9 | Boot time script file | 49 |
| 8.1 | Router Restful API Development Standard | 52 |

Chapter 1

Introduction

1.1 Motivation

Significant growth in IoT and Home Automation in developed and developing nations opens market for huge potential products. HUB acts as a main gateway for Home Automation products. Modular router bring additional functionality to use the Home Automation HUB as smart router by eliminating the traditional non-modular router.

1.2 Objective

The objective behind this project is to provide custom modular router to fulfill various requirements in smart home environment such as providing internet connectivity to end things, home users.

1.3 Scope

This project can be used for fast and secure router which is also used as hub in t-home.

1.4 Requirements

For this project at Smartron, requires knowledge of Linux platforms, networking tools like DHCP, HostAPD, IPtables, wpa_supplicant. For writing modules for web user interface for this project, requires knowledge of nodejs script language.

1.5 Gantt Chart

The timeline of project work from the start of the project is shown in below gantt chart.

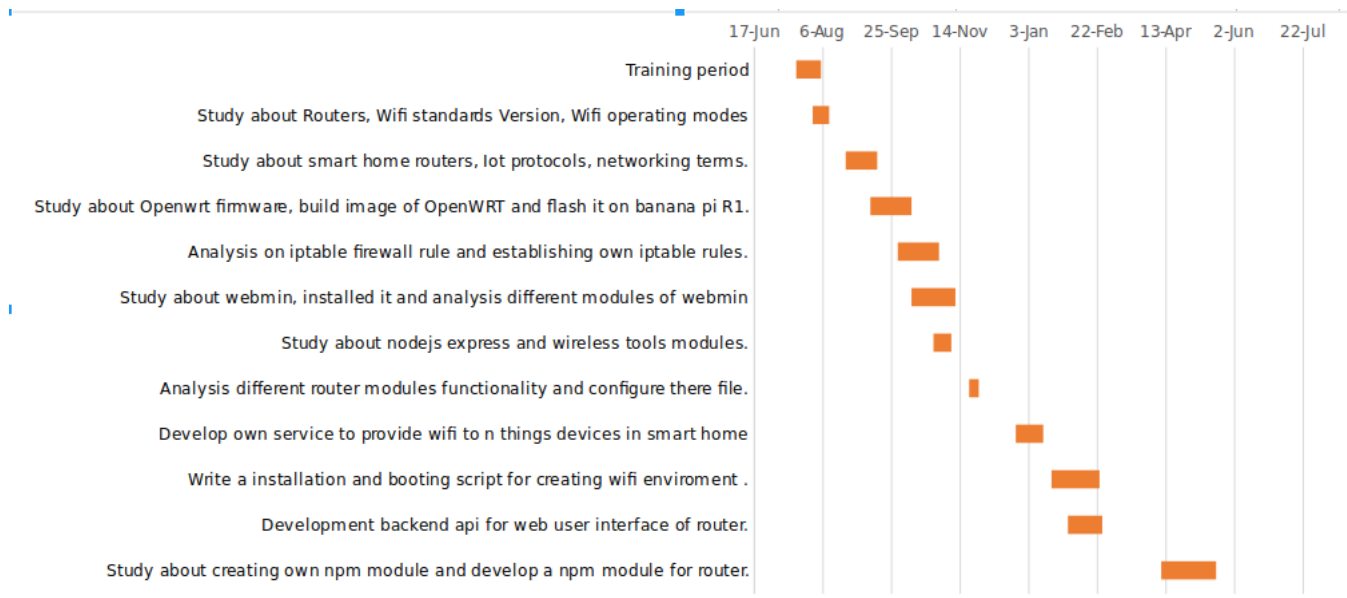


Figure 1.1: Gantt Chart

1.6 Router Introduction

For home automation , wifi router is essential so that you can interact with your home automation devices while you are away from home, In home automation we

have more than one smart devices or computers for watching live streaming video or voip calls and all require wifi router to communicate with user or each other. Wifi router is essential to control all smart appliances and more.

Wifi standard play a major role for speed or connection of wifi in router or devices connected to router. Wifi standard are assigned to 802.11 and with its various version a, b, g, n, ac. This various version of wifi are both forward and backward compatible , so any device with this wifi version work with both forward and backend version. 802.11g is the most obsolete version.

All wireless communication are done on some frequency band, this frequency band work like roads to travel data on devices. Wifi standard 802.11g based devices are working on 2.4ghz frequency band. Wifi standard 802.11n based devices have support of both 2.4ghz and 5ghz frequency band. As we see 2.4 ghz band is too much over crowded due to lot of devices are connected to it and 5ghz band is less crowded so in 802.11n wifi standard it is added to reduce traffic. The devices which are connected to both 2.4ghz and 5 ghz band are called dual band devices. So all the newly invented wifi standard 802.11n and ac have dual band support for its devices. But there is also another major drawback while switching to higher frequency band because if any obstacle came in higher frequency band than there is major lose occur rather than lower frequency band.

In AC wifi standard these issue is also resolved by beamforming that it is aim to resolve the range problem by transmitting signal only in the direction of connecting device rather than transmitting it in all direction of connection. Usually a single directional signal have higher range rather than omnidirectional signal.

There are two type of dual band device, the router which allow you to select on which band you want to transmit your data is either on 2.4ghz or 5ghz, and the other type is which allow you to transmit data on both 2.4ghz and 5ghz device simultaneously. All the live streaming video and games use the 5ghz band for working without latency because 5ghz band have less traffic than 2.4ghz band.

Router quality of services is called as traffic shapper , it prevent unequal distribution

of bandwidth , It assign priority to each service and device on network and according to user requirement you can set bandwidth for the service like for live streaming video you can provide more bandwidth and for downloading purpose less, it all depends on user requirement. Router QOS is very cumbersome process , it allow us to set the downloading and uploading speed which your Internet service provider support with knowledge of protocols. Router provide automated QOS management via WMM(wifi multimedia), It provide a prioritization from highest to lowest in which : voice , video, most traffic from app rather than voice and video , background downloading. Wifi channel in each frequency band are used to make data flow smooth and make traffic problem less , its like a lane in highway road for resolving traffic problem. The 2.4 Ghz band has 11 channel of 20Mhz wide which are available for wireless device. But there is overlapping occur between multiple channel, to resolve these we always have to stick to particular channels which are not undergoes overlapping channels. The maximum non overlapping channels are 1, 6, and 11 ,these three channels are used maximum time in each wireless interface.

For removing these overlapping channel , it is always recommended to use 802.11n, ac based wireless device which have dual band frequency channel which have less overlapping channel.

Tools used are Banana Pi R1, IPTable , DHCP, HostAPD, DNS, Webmin, DIR-615 Router. Bananan Pi R1 is open hardware router and it run on different open source operating system like android, bananian and openWRT.It has 300Mbs wireless N capabilities, 1 gigabit WAN port, 4 gigabit LAN ports . It has 1gb DDR3 memory and open source platform. IPTables has different set of rules for security in linux kernel. Each table has user defined chain and built in chain. IPTable is used to maintain the table of iptable packets rule. Each chain has set of rules in it which can match with set of packets and follow that rule. IPTable rule has specific criteria for each set of packet and target and if it does not matches than than next IP table rule is applied. DHCP(Dynamic host configuration protocol) is dynamic distribute the network parameter like IP addresses for service and interfaces. DHCP enable user to

request network parameter like IP address automatically and reducing the manual configuration setting. DNS (Domain name system) mapped the domain name to IP address and DNS server has record of ip address and host name. Webmin is web based interface for system management for unix.

1.7 Feature to increase speed of wireless router

There are some points which you have to notice when buying a wireless router for its speed.

1.7.1 Flash memory

All the drivers, web interface and files information is store in flash memory in binary form in router. So as fast our flash memory is than its response in router is also fast and our router perform faster.As fast our flash memory is decide how fast it load all information in its RAM.

There are various flash memory which is MLC flash and SLC flash.

1.7.2 RAM

CPU used temporary information which is stored in RAM memory. For example in flash memory we watch a picture which is stored in it.CPU take the picture from flash memory bit by bit and stores it in RAM memory. For showing picture they translate a binary code in a way that monitor will understand,the CPU translate instruction to monitor which is stored in RAM. RAM and CPU decide the connection speed.

1.7.3 Radios

Higher data rate can transmit through 5GHz band and it have less traffic than 2.4GHz band, whenever we have to transmit higher data rate we will transmit

through 5GHz band. It has high bandwidth with less traffic. AC and 802.11n will always transmit through less traffic band.

1.7.4 CPU

Router perform its task with the help of CPU, CPU will direct all the activity or instruction to go to its peripheral and perform its task of router function. As our CPU is better our router output is also better. Multi-core CPU is much better because at a time it can perform multiple task.

Chapter 2

Literature review

2.1 Router

Significant growth in IoT and Home Automation in developed and developing nations opens market for huge potential products. HUB acts as a main gateway for Home Automation products. Modular router bring additional functionality to use the Home Automation HUB as smart router by eliminating the traditional non modular router. Smart home are not efficient without wifi router because it control all the appliance activity in smart home. thome router is a modular router which have dependency on tools like hostapd, dhcp, wpa_supplicant, iptable firewall etc. Router is a gateway which flow your data on your network and internet. Its job is to communicate between internal local network and outer network(internet).

2.2 Analysis on Router

As we see smart router nowadays not only used for routing of packets but also many devices are connected to router for forwarding of data like Bluetooth speakers, TV and many other so it require more security, management, better GUI(graphical user interface),USB storage. Smart Router are used in smart home and work like hub which connect different home appliance. There are various company which come

with their own smart router and have their own features.

Almond+

With a touch screen shown in figure 2.1 and the ability to connect with household automation appliances, the Securifi Almond+ can not only fill a home with Wi-Fi but control it as well. With ZigBee and Z-Wave antennas, this router lets you connect to and control hundreds of smart home devices, from light bulbs to garage door openers, right from your smartphone [1]. It also has a 3.5 inch touch screen display with no antenna in its router box. It has a 4*2 grid icons on its display screen. In the back of the router are four Gigabit wired LAN ports as well as a pair of USB 3.0 ports that work with hard drives and printers[1]. We can configure the router by tapping on its screen on its router. Almond+ comes with an AC adapter, a flat Ethernet cable and mounting hardware, but no CD[1]. After its configuration and it connects to internet we can select its range extender, access point.

Chime

Chime is a cloud based service architecture shown in figure 2.2 which makes it easy to implement and give extra profit to market and it easily integrates with low to high end routers and works on every router platform like ARM, Qualcomm[2]. It provides online single security service to the whole home. Parents have all the access with network in Chime. It provides a chat based user interface which is easy to use and also has fun, it also provides safety to its third party user through its cloud based service architecture. Chime was created in the Innovation Labs of AVG Technologies[2]. From Chime you can get an anti-virus for home for security purposes. It provides mesh networking topology for wireless connection. You can control internet through its Chime application. Chime is a dual band router. It has a powerful dual core CPU, 2*1GB ethernet port, 1 USB 2.0 port, bluetooth, 802.11 ac router[2]. Its anti-virus software protects



Figure 2.1: Almond touch screen display[1]

from malware attack. It also do content filtering and provide parental control. It is equipped with VPN and give your online connect more private. And it is wall mountable also.

Torch

Torch lets you create a profile for each child, and a separate profile for the adults, Drag and drop devices to the profiles of those who use them, so you're no longer having to play device manager, And we know each child is different, so each child's settings can be different too, As a bonus, Torch provides real, human tech support for any tech questions you may run into[3]. Taking fast to the next level, the Torch router comes equipped with the latest 3 Stream 802.11 ac technology, With a Gigabit LAN and WAN port, and 6 smart antennas, the Torch router will keep all your devices

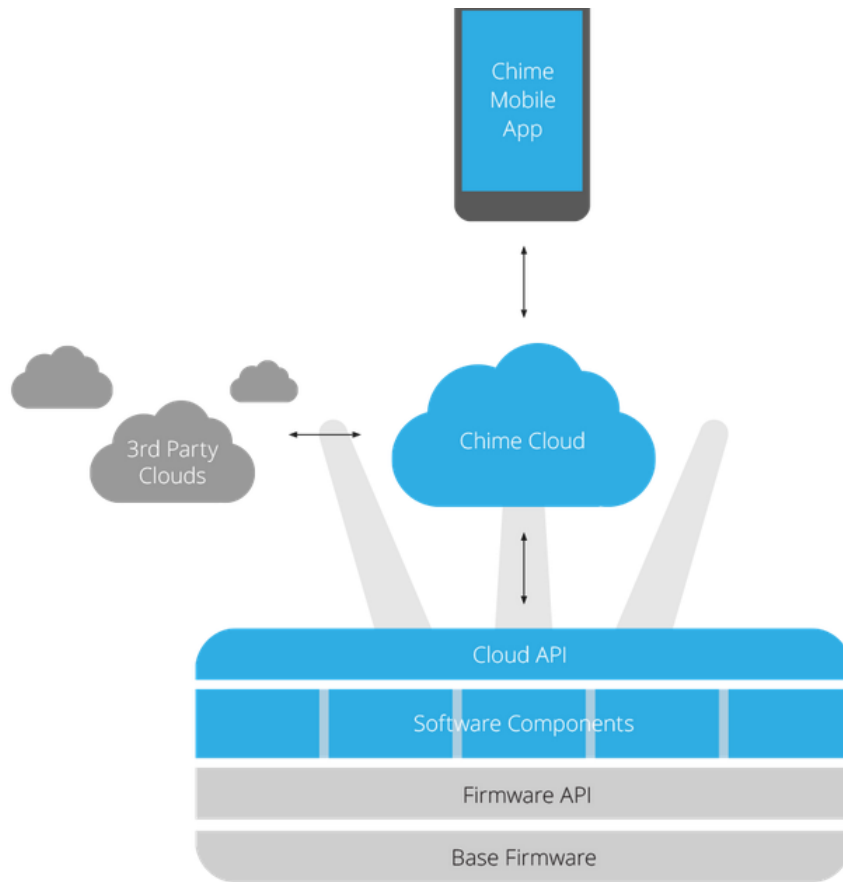


Figure 2.2: Cloud service architecture in chime[2]

humming along with optimum speed and power, It have features like it have quad band, USB 2.0 port,12v 2.0amp dc power input[3].It is very easy for parent to control their child activity through torch.

Eero

With eero, you simply plug one device into your modem, Additional eeros simply need power from a standard outlet, They automatically connect to each other to create a single wireless mesh network that covers your whole home, If you already have Ethernet wiring, you can always choose to hardwire your additional eeros [4].

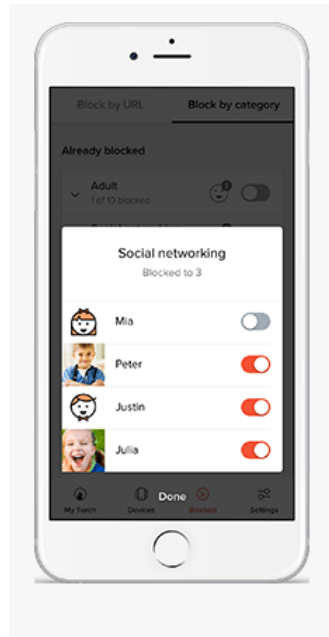


Figure 2.3: Torch Application parental control feature[3]

It automatically update its software, and provide best software for security purpose, manage from remote device , you can set screen time for display and for managing internet access[4].Figure 2.3 show the remote application for parental control of torch.

Luma

Luma provide secured internet connection with fast speed of internet, you can control your device from any remote device by simply installing its app, it also provide time management system ,you can pause the internet by simply clicking on button on your remote device for any work, it will help you in black listing any service or IPaddress, you can make different profile for different user and filter whatever site you want to filter, you can easily give guest access without any actual password of your router, you can make different account for guest access.

2.3 Router feature

These are the following feature of Smart Router:

Parental Control: Parents can control the access to internet by their child during night time ,they directly pause the internet or they can set time for internet access or they can block certain services

Media Prioritization: In smart router they can prioritize which media get more bandwidth which less, for example for live streaming video we can give higher bandwidth, for games we can higher bandwidth and for downloading purpose less bandwidth.

Black Listing: We can block the illegal sites or IPaddress, We can also block social sites in office

High End Security:We can provide security by installing secure software or by installing antivirus for protecting it from malware attack or phishing attack.

GUI: Provide conversational UI(user interface)

Remote Control: Control router from remote device by installing its app.

Mesh network: Mesh network provide strong wireless network

Time management: Time is decided for each user for internet access.

2.3.1 Router QOS

Beamforming: It is a technology send all signal to end user rather than broadcasting in all direction.When router broadcast a signal, it broadcast the data in all directions. With beamforming the router determines where your device is located and project a stronger signal in specification.

Quad band: Router support both frequency 2.4Ghz and 5Ghz. Frequency with 2.4ghz have been used for downloading purpose or live streaming purpose and have long range with less interference.Frequency with 5Ghz used for online gaming purpose but have short range with less interference.

You can prioritize bandwidth from high, medium, low for each device.Below figure 2.4 show the prioritize bandwidth from high to normal for each device.

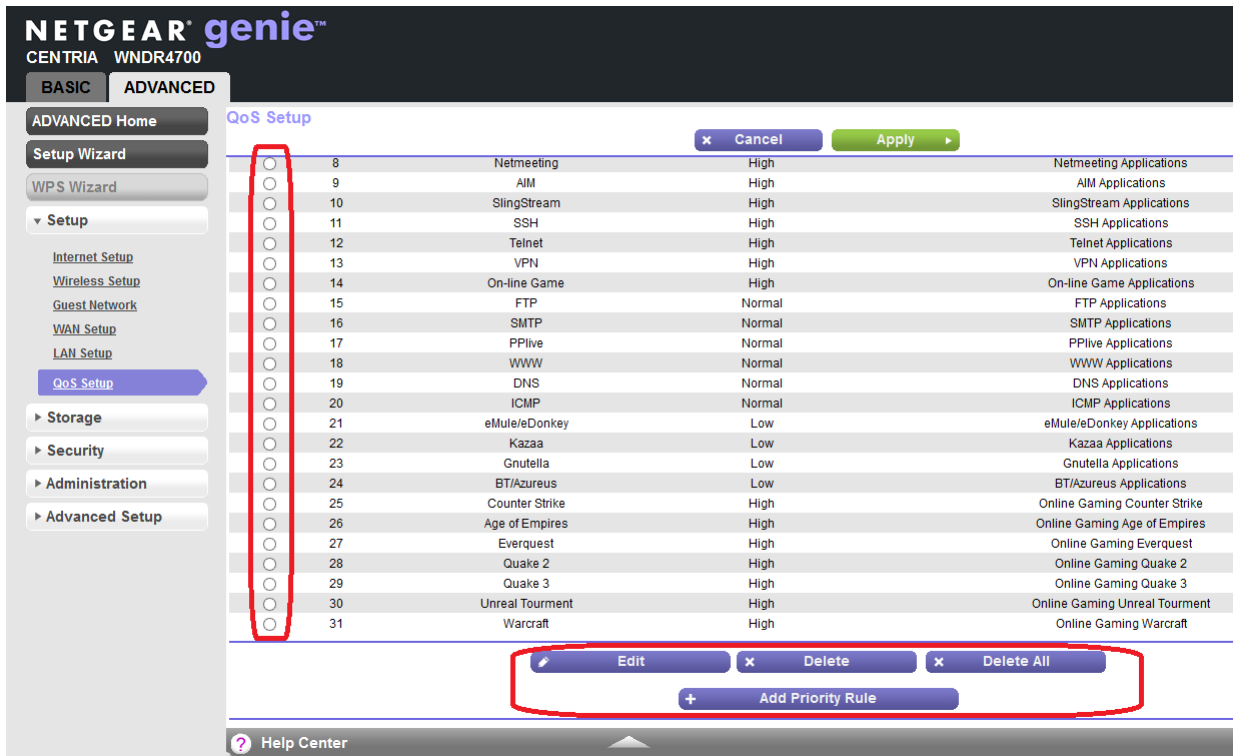


Figure 2.4: Netgear router QOS[15]

Chapter 3

Tools of Router

There are various tools in smart router DHCP, DNS, HostAPD,IW, WPA_SUPPLICANT, WPA_CLI, IPtables:

3.0.1 Dynamic Host Configuration Protocol

Dynamic Host Configuration Protocol (DHCP) is a client/server protocol that automatically provides an Internet Protocol (IP) host with its IP address and other related configuration information such as the subnet mask and default gateway, It is communication protocol that assign a IPaddress, default gateway, Primary DNS, Secondary DNS, subnet mask to user, Every device on a TCP/IP-based network must have a unique unicast IP address to access the network and its resources, Without DHCP, IP addresses for new computers or computers that are moved from one subnet to another must be configured manually; IP addresses for computers that are removed from the network must be manually reclaimed [16].

DHCP will automatically assign IPaddress to host computer by using DHCP server.DHCP Server provide IPaddress on lease to DHCP enable client and maintain a DHCP pool of certain range of IP's , whenever a client request for IP from DHCP server than it will give IP within this range of IP. IPaddress are assigned in two way: statically (permanent) or dynamically (lease),Permanent IPaddress is assign to single client

which cannot be changed and Lease IP address is assigned to particular client for a fixed period of time. When this time period gets over, the client again claims DHCP server for another IP address. Pros of DHCP server is it requires less administration and provides flexible configuration [16]. Below figure 3.1 shows the cycle of DHCP client and server IP address request response flow.

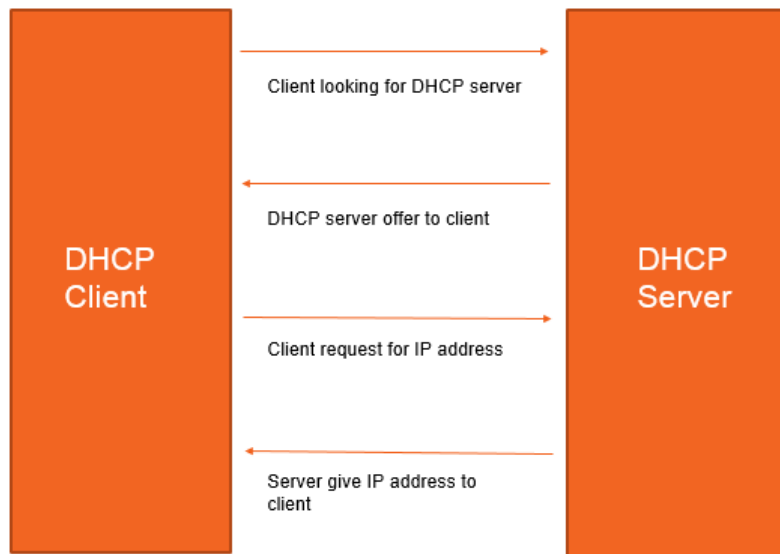


Figure 3.1: DHCP Flow

3.0.2 Domain Name System

When in the past there was no DNS, people used to download a file having all information related to host name and their IP address. As the number of hostnames increases, it becomes impossible to set records. After that, the domain name system was introduced.

IP address is a 32-bit number which is difficult to remember rather than a domain name. In the domain name system, mapping of domain names to each IP address is done. In reverse domain

name assigning IP address to domain name.

DNS server has records of all local area network, IP address and host name. Domain name system give a way to match the current domain name with all the recorded domain name. It is very typical to remember IP address of all your favourite site, domain name server have record of all these domain name and IP address.

There are three type of DNS server:

Primary server:

Primary server contain information all about its zone file.

Secondary server:

It transfer information about a zone file from some primary server. But it cannot modify or update any zone file.

root server:

Root server give the authority to other domain server to complete task or store zone file.

DNS server store the file information related to domain name and its ip address.

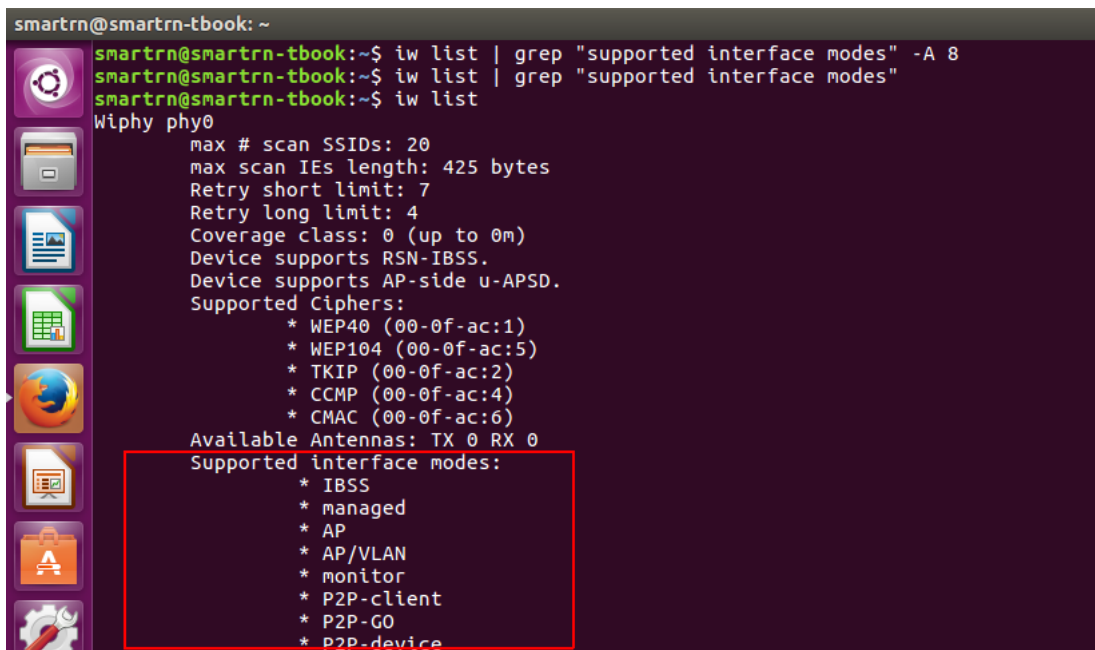
3.0.3 HostAPD

Hostapd (Host access point daemon) is a user space software access point capable of turning normal network interface cards into access points and authentication servers. The current version supports Linux (Host AP, madwifi, mac80211-based drivers) and FreeBSD (net80211)[17].

For creating software access point we require our wireless card have support of access point mode.

Command used for checking Access point mode support is there in your wireless card in ubuntu based system is

```
$ iw list
```



```
smartrn@smartrn-tbook: ~
smartrn@smartrn-tbook:~$ iw list | grep "supported interface modes" -A 8
smartrn@smartrn-tbook:~$ iw list | grep "supported interface modes"
smartrn@smartrn-tbook:~$ iw list
Wiphy phy0
  max # scan SSIDs: 20
  max scan IEs length: 425 bytes
  Retry short limit: 7
  Retry long limit: 4
  Coverage class: 0 (up to 0m)
  Device supports RSN-IBSS.
  Device supports AP-side u-APSD.
  Supported Ciphers:
    * WEP40 (00-0f-ac:1)
    * WEP104 (00-0f-ac:5)
    * TKIP (00-0f-ac:2)
    * CCMP (00-0f-ac:4)
    * CMAC (00-0f-ac:6)
  Available Antennas: TX 0 RX 0
  Supported interface modes:
    * IBSS
    * managed
    * AP
    * AP/VLAN
    * monitor
    * P2P-client
    * P2P-GO
    * P2P-device
```

Figure 3.2: Supported interface mode

Above figure 3.2 show the iw list command give details for all wireless network interface card and there supported operating mode. If there is AP in interface support

mode in iw list than your system have access point mode support with hostapd. Access point is like a wireless switch, Access point can use one radio band at a time either 2.4ghz or 5 ghz. So it create one access point at 2.4ghz and other at 5ghz, so it is dual band AP. An access point having 2.4 ghz of band support have hardware support of b,g and n at same time and if it have support of 5gh band than at same time it have support of a,n and ac. An access point have support of multiple ssid but all will have same band AND channel.

Hostapd has ability to create more than one access point on same wireless interface card, and it support upto 5-6 access point on same wireless card. It create multiple access point on different interface card all in single occurrence of hostapd. It uses two radio band 2.4 or 5 ghz on same interface card but for these feature it require card with two radio wave support. But it cannot build multiple software access point on different channels within same network interface card because multiple access point within same interface card only share same channel. By using dhcp server support access point can distribute ipaddress to the device connecting to it but assigning ipaddress to access point is not a work of hostapd.

Table 3.1: Frequency Band [17]

| Frequency | 802.11 | channels | Interferences |
|-----------|--------|-----------|-------------------------------------|
| 2.4GHz | b/g/n | 11 to 14 | high because channels overlap |
| 5GHz | a/n/ac | around 20 | low because channels do not overlap |

Above table 3.1 give details for frequency band and there supported wifi IEEE802.11e standard versions.

Hostapd minimal configuration

```
interface= wlan0
```

```
driver= nl80211
```

```
ssid= smartron
```

```
channel= 1
```

Before configuring hostapd we always need to have information regarding channel, driver, encryption mode, wpa mode support, hardware support.

Table 3.2: 802.11 [17]

| Technology | Band | Max Speed | notes |
|------------|-------------|-----------|--|
| 802.11a | 5Ghz | 54Mbps | obsolete |
| 802.11b | 2.4Ghz | 11Mbps | obsolete |
| 802.11g | 2.4Ghz | 54Mbps | on its way to become obsolete |
| 802.11n | 2.4 or 5Ghz | 600Mbps | a device can only use one band at a time |
| 802.11ac | 5Ghz | 6777Mbps | on its way to become very popular |

Above table 3.2 give details of IEEE802.11 versions.

Interface Interface is used in hostapd configuration to describe which wireless interface to use.

hw_mode hw_mode is used to lie the operating mode of network interface and give channel that have support that is valid for hardware.

Channel Channel always lie in hostapd according to mode set in hw_mode and country wireless regulatory rules.

ssid Service set Identifier is used to recognize the access point by client in there region.

driver Network interface card for computer require software to make them function and this is done by driver. For example iwlwifi is driver which is manufactured by intel and it have cfg80211 support and it also support access point mode and it have support of physical mode a,b,g,n,ac and it support bus PCI-E.

Command in ubuntu to check which network driver is there in your system is:

```

smartrn@smartrn-tbook:~$ lspci -k | grep -A 3 -i "network"
01:00.0 Network controller: Intel Corporation Wireless 7260 (rev bb)
Subsystem: Intel Corporation Dual Band Wireless-AC 7260
Kernel driver in use: iwlwifi
Kernel modules: iwlwifi
smartrn@smartrn-tbook:~$ modinfo iwlwifi | grep 'depend'
depends:         cfg80211
smartrn@smartrn-tbook:~$ █

```

Figure 3.3: Network Driver Details[2]

Above figure 3.3 give the network driver details.

Example of driver are:

cfg80211: cfg80211 is the kernal configuration management for wireless devices. cfg80211 are under development.

nl80211: nl80211 is user side configuration for wireless devices.

nl80211 are under development. Wireless network interface work in one operating mode, here are the all the listed operating modes of wireless network interface-

Access Point: An Access Point acts as the Master device in a managed wireless network, It holds the network together by managing and maintaining lists of associated Stations, It also manages security policies, The network is named after the MAC-Address (BSSID) of the AP, The human readable name for the network, the SSID, is also set by the AP, To use access point mode in linux you need to use hostapd[19].

Station infrastructure mode:The Station device connects to an access point by sending certain management packets to it, This process is called the authentication

and association, After the AP sent the successful association reply, the station is part of the network, This mode is also called managed in the wireless extension tools[19].

Monitor mode: Monitor mode is a passive-only mode, no packets are transmitted, All incoming packets are handed over to the host computer completely unfiltered, This mode is useful to see what's going on on the network[19].

With mac80211, it is possible to have a network device in monitor mode in addition to a regular device, this is useful to observe the network whilst using it, However, not all hardware fully supports this as not all hardware can be configured to show all packets while in one of the other operating modes, monitor mode interfaces always work on a best effort basis[19].

With mac80211, it's also possible to transmit packets in monitor mode, which is known as packet injection, This is useful for applications that wish to implement MLME work in userspace, for example to support nonstandard MAC extensions of IEEE 802.11[19].

Ad-Hoc (IBSS) mode: The Ad-Hoc mode is used to create a wireless network without the need of having a Master Access Point in the network, Each station in an IBSS network is managing the network itself[19]. Ad-Hoc is useful for connecting two or more computers to each other when no (useful) AP is around for this purpose[19].

Wireless Distribution System : The Distribution System is the wired uplink connection to an AP[19]. The Wireless Distribution System is the wireless equivalent to it, WDS serves as a wireless communication path between cooperating APs, it can be used instead of cabling[19].

Mesh: Mesh interfaces are used to allow multiple devices to communication with each other by establishing intelligent routes between each other dynamically[19].

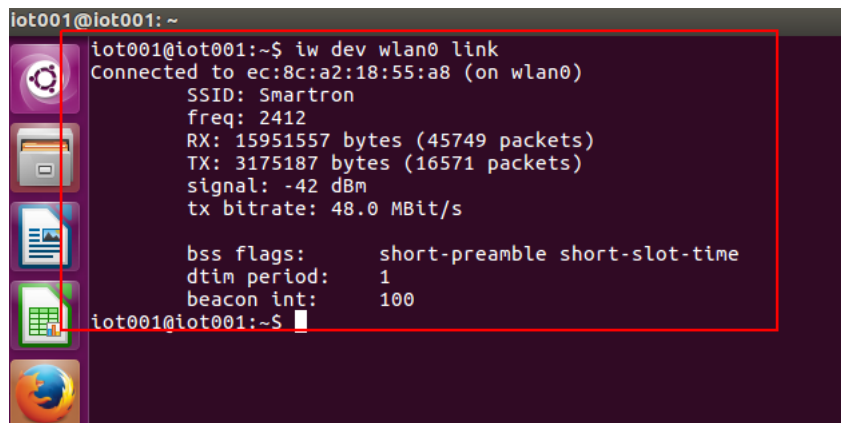
Authentication: Authentication can be done in many way in hostapd. There are points related to authentication in hostapd given below-

WPA: Wifi protected access is mainly used for security of wireless network. It

exchange the WEP(Wired Equivalent Privacy) wifi standard, it provide more encryption of data as compared to wep and also provide authentication done by user.It provide sophisticated authentication based on EAP(extensible authentication protocol) and 802.1x and it have authentication server RADIUS to authenticate user. Access point can create on both wpa/wep mode to support both wpa and wep client and home user are only connected to wpa based access point during authentication process they will give password and tkip(Temporal Key Integrity Protocol) encryption.

WPA2: WPA2 uses the Counter Mode Cipher Block Chaining Message Authentication Code Protocol (CCMP) and it is compatible with ieee 802.11i. It is more compatible and give strong authentication than tkip and wpa.

3.0.4 IW



```
iot001@iot001: ~  
iot001@iot001:~$ iw dev wlan0 link  
Connected to ec:8c:a2:18:55:a8 (on wlan0)  
SSID: Smartron  
freq: 2412  
RX: 15951557 bytes (45749 packets)  
TX: 3175187 bytes (16571 packets)  
signal: -42 dBm  
tx bitrate: 48.0 MBit/s  
  
bss flags:      short-preamble short-slot-time  
dtim period:   1  
beacon int:    100  
iot001@iot001:~$
```

Figure 3.4: iw_list

IW uses wireless extension interface which is new nl80211 based cli configuration for wireless devices and it support all new drivers that have been added to kernal, it replace iwconfig and mostly recommend to use iw for wireless devices. It have

different feature like iwlist which provide wireless device information about access point address, ssid, frequency, operating mode, channel, security, signal.

To get the information about client which are connected to your software access point you can use IW. Below figure 3.5 give the command used in ubuntu to get the number of client connected to our software access point.

3.0.5 wpa_supplicant

wpa_supplicant is cross platform supplicant with support for wep, wpa and wpa2(ieee 802.11), it is suitable for desktop, laptops and embedded systems [20].

It is background program that is designed as daemon program and work like backend component controlling wireless network, It support wpa_cli which is separated front-end program ,wpa_cli is text based program for interacting with wpa_supplicant, wpa_supplicant is the IEEE 802.1X/WPA component that is used in the client stations,It implements key negotiation with a WPA authenticator and it controls the roaming and IEEE 802.11 authentication/association of the wireless driver [20].The first step to connect to an encrypted wireless network is having wpa_supplicant obtain authentication from a WPA authenticator[20]. In order to do this, wpa_supplicant must be configured so that it will be able to submit the correct credentials to the authenticator,Once the authentication is successful, it will be possible to connect to the network by obtaining an IP address in the usual way[20].

3.0.6 wpa_cli

wpa_cli is a textbased frontend program for interacting with wpa_supplicant, It is used to query current status, change configuration, trigger events, and request interactive user input, wpa_cli can show the current authentication status, selected security mode, dot11 and dot1x MIBs, etc , In addition, it can configure some variables like EAPOL state machine parameters and trigger events like reassociation and IEEE 802.1X logoff/logon, wpa_cli provides a user interface to request authentication information, like username and password, if these are not included in the configuration, This can be used to implement, e.g., one-time-passwords or generic token card authentication where the authentication is based on a challenge-response that uses an external device for generating the response, The control interface of wpa_supplicant can be configured to allow non-root user access (ctrl_interface GROUPparameter in the configuration file)[21].

This makes it possible to run wpa_cli with a normal user account, wpa_cli supports two modes: interactive and command line[21]. Both modes share the same command set and the main difference is in interactive mode providing access to unsolicited messages (event messages, usernamepassword requests)[21].

Interactive mode is started when wpa_cli is executed without including the command as a command line parameter, Commands are then entered on the wpa_cli prompt, In command line mode, the same commands are entered as command line arguments for wpa_cli[21].

When wpa_supplicant need authentication parameters, like username and password, which are not present in the configuration file, it sends a request message to all attached frontend programs, e.g., wpa_cli in interactive mode[21].

```

TDL S peer:      no
!:.telnuc@intelnuc-desktop:~$ iw dev wlp2s0 station dump
Station e0:98:61:24:f0:d1 (on wlp2s0)
inactive time: 13300 ms
rx bytes: 128979
rx packets: 808
tx bytes: 159514
tx packets: 483
tx retries: 22
tx failed: 0
signal: -32 [-38, -32] dBm
signal avg: -32 [-38, -33] dBm
tx bitrate: 11.0 MBit/s
rx bitrate: 1.0 MBit/s
authorized: yes
authenticated: yes
preamble: long
WMM/WME: no
MFP: no
TDL S peer:      no
Station 58:91:cf:c3:07:da (on wlp2s0)
inactive time: 32 ms
rx bytes: 192861
rx packets: 719
tx bytes: 240370
tx packets: 381
tx retries: 278
tx failed: 0
signal: -66 [-79, -66] dBm
signal avg: -66 [-72, -68] dBm
tx bitrate: 11.0 MBit/s
rx bitrate: 11.0 MBit/s
authorized: yes
authenticated: yes
preamble: long
WMM/WME: no
MFP: no
TDL S peer:      no
Station 74:e2:8c:d3:2c:7c (on wlp2s0)
inactive time: 0 ms
rx bytes: 67742
rx packets: 103
tx bytes: 14966
tx packets: 64
tx retries: 7
tx failed: 0
signal: -64 [-64, -72] dBm
signal avg: -56 [-67, -58] dBm
tx bitrate: 11.0 MBit/s
rx bitrate: 2.0 MBit/s
authorized: yes
authenticated: yes
preamble: short
WMM/WME: no
MFP: no

```

Figure 3.5: iw_list

Chapter 4

OpenWRT Framework

4.1 Introduction

OpenWRT is linux based firmware embedded operating system, It is free and Open source, it is fully writable file system with package management and you can choose your own package to suit your application of device and make you independent from choosing vendor configuration and application, for developer it is a framework to build an application without building the whole firmware [18].

It's latest version is OpenWRT Chaos Calmer 15.05.1, it come with linux kernal update version and more security fixes,driver updates, support new devices, package fixes, OpenWRT provide table of hardware which give the list of router which is compatible with OpenWRT and its version, Supported device in openWRT give the full description of how to install openWRT with its latest version which is suitable[18].

There are pre-built snapshot which have install image of openWRT but not contain the installed web-interface LUCI.And whenever we are installing the packages for our platform, we always have to check the date for it[18].

4.2 OpenWRT Features

OpenWRT is free and Open source firmware

It has built in package manager, OPKG , default user interface is CLI, WebUI's, large repository of packages, command line access via SSH or PUTTY, Real time networking monitoring, Allow to setup Dynamic DNS, Mesh networking setup, Provide DHCP on lease, busybox and Uclibc, OpenWRT works on linux, BSD, and MAC operating system[18].

4.3 Building an Image

Steps to build image of OpenWRT chaos calmer 15.05.1 latest version image for lamobo banana Pi-R1 open wireless router.

Step1: Install git to download the OpenWrt source code, and build tools to do the cross-compilation process:

- `sudo apt-get update`[6]
- `sudo apt-get install git-core build-essential libssl-dev libncurses5-dev unzip gawk`[6]
- `sudo apt-get install subversion mercurial`[6]

Step2 : Getting trunk source of OpenWRT:

- `git clone git://git.openwrt.org/openwrt.git`[6]

Step3 : To make the standard packages as well as your custom feed available in make menuconfig:

- `cd /openwrt`
- `./scripts/feeds update -a`
- `./scripts/feeds install -a`

Step4 : Configure OpenWRT for banana Pi R1 using make menuconfig and we have to select this option for support of banana Pi R1 is

- make prereq && make defconfig && make menuconfig [6]
- Target System - Allwinner A1x/A20/A3x [6]
- Target Profile - Lamobo R1 [6]

Step5 : Build OpenWRT

- make

Step6 : Obtain final openWRT image

- cd ~/openwrt/openwrt/bin/sunxi [6]

Step7: Dump this image on banana Pi R1 SD card:

- dd if=./openwrt-sunxi-Lamobo_R1-sdcard-vfat-ext4img of=/dev/your_sd_reader bs=1m; syncopenwrt-sunxi-Lamobo_R1-sdcard-vfat-ext4img [6]

4.4 Banana Pi R1

Banana Pi R1 is open hardware router, that can run on a variety of open source operating systems including OpenWrt, Android, and Bananian. It also has 4 Gigabit LAN ports, 1 Gigabit WAN, and 300Mbs wireless N capabilities and Banana Pi R1 is an open platform device, it is for anyone who wants to play and build with developer technology instead of simply using consumer technology[5], Below figure 4.1 show the Banana Pi R1 chip.

Feature: Dual-core 1.0GHz CPU, 1 GB DDR3 memory, Mali-400 MP2 with Open GL ES 2.0/1.1[5].



Figure 4.1: Banana Pi R1 image [14]

which have to match set of packets and this rule which matches has to decide what to do with packets. Above figure 5.1 show the details of chain process for transmitting packets. IPtables rule have some criteria for each packet and target if it does not matches than it will inspects some other rule.

5.2 Tables

There are three predefined or independent tables which are:

Filter: It is a default table in IPtables, If you are not using your own table than you are using filter table[8]. It has following three built in chain which are:

INPUT chain: IPpackets coming into local server or incoming the IPpackets to firewall[8].

FORWARD chain: IPpackets routed through local server[8].

OUTPUT chain: Out going from firewall or going out of local server[8].

NAT: This table is confer when new connection is established for packets and rewriting the source and destination address of packets as they pass through firewall or router[8]. It is used for destination network address translation[8].It have three built-in chain PREROUTING, OUTPUT and POSTROUTING:

PREROUTING chain: Alters packets before routing. i.e Packet translation happens immediately after the packet comes to the system (and before routing)[8]. This helps to translate the destination IP address of the packets to something that matches the routing on the local server[8]. This is used for DNAT (destination NAT)[8].

OUTPUT chain: NAT for locally generated packets on the firewall[8].

POSTROUTING chain: Alters packets after routing. i.e Packet translation happens when the packets are leaving the system. This helps to translate the source ip address of the packets to something that might match the routing on the destination server[8]. This is used for SNAT (source NAT)[8].

Mangle: Mangle is for modifying packet in firewall and it alter QOS in TCP header[8]. It have five built in chain PREROUTING(DNAT),OUTPUT, INPUT, FORWARD, POSTROUTING(SNAT)[8].

Raw: Iptables Raw table is for configuration exceptions, Raw table has the following built-in chains:PREROUTING chain,OUTPUT chain[8].

5.3 Targets and Jumps

ACCEPT: Rule is accepted and no further processing is required [7].

REJECT: Drop the packet, no further processing[7].

DROP: Drop the packet and send error message to the host[7].

LOG : Log all packet that matches the rule,Since the packets are logged by the kernel, the /etc/syslog.conf file determines where these log entries are written [7].

QUEUE: The packet is queued manage by application[7].

DNAT: Again writing the destination address.

SNAT: Again writing the source address

MASQUERADE: Source IP address is similar to firewall interface.

5.4 IPtables Switch Commands

-m conntrack: iptables has a set of core functionality, but also has a set of extensions or modules that provide extra capabilities[9].

-ctstate: This is one of the commands made available by calling the conntrack module[9]. This command allows us to match packets based on how they are related to packets we've seen before,ESTABLISHED to allow packets that are part of an existing connection[9]. We pass it the value of RELATED to allow packets that are associated with an established connection, NEW to allow packet with new connection, INVALID the packet cannot be identified[9].

-m multiport -sports <port, port >

Table 5.1: IPtables switch commands [7]

| Commands | Description |
|----------|---|
| -t | table |
| -j | jump to target |
| -n | numeric details |
| -s | source IPaddress |
| -d | destination IPaddress |
| -i | input interface |
| -o | output interface |
| -sport | source port number |
| -dport | destination port number |
| -v | verbose |
| -x | exact value of packets and byte counters |
| -L | List of all chain in table |
| -X | For delete chain |
| -D | For deleting all the rule from chain |
| -I | For inserting rule in chain |
| -N | For new chain |
| -S | To see current rule |
| -P | All the default Policy |
| -F | This command use to flush all rule from chain |

Above table 5.1 give the details of generalize command in IPtable rule and there description.

Generalize command

-A: append one or more rule in chain

-m multiport -dports <port, port >

-m - state <state>: state : ESTABLISHED, NEW, RELATED, INVALID

5.5 IPtables rule

To install IPtable in linux: `sudo apt-get install iptables` [9]

For listing the iptables chain rule: `iptables -L -n -v`[9]

default chain policies :

```
iptables -P INPUT DROP[9]
```

```
iptables -P FORWARD DROP[9]
```

```
iptables -P OUTPUT DROP[9]
```

Delete existing rules: `iptables -F` [9]

Block a specific ip address: `iptables -A INPUT -s 10.11.15.60 -j DROP`

To block only TCP traffic on eth0 connection for this ip-address:

```
iptables -A INPUT -i eth0 -s 10.11.15.60 -j DROP
```

```
iptables -A INPUT -i eth0 -p tcp -s 10.11.15.60 -j DROP
```

To insert a rule from a chain: `iptables -I INPUT 1 -i lo -j ACCEPT`[9]

To delete a rule from a chain: `iptables -D INPUT rule number`[9]

Flush a single chain: `iptables -F INPUT`[9]

Rule to make a user defined chain: `iptables -N mytable` [9]

ICMP echo-request type will be block by:`iptables -A OUTPUT -p icmp --icmp-type echo-request -j DROP`[9]

Enable or allow ICMP ping incoming client request:

```
iptables -A INPUT -p icmp --icmp-type 8 -s 0/0 -d SERVER_IP -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT[9]
```

```
iptables -A INPUT -p icmp --icmp-type 0 -s SERVER_IP -d 0/0 -m state --state ESTABLISHED,RELATED -j ACCEPT[9]
```

ICMP echo-request type will be block by: `iptables -A OUTPUT -p icmp --icmp-type echo-request -j DROP`[9]

Allow traffic on loopback:

```
iptables -A INPUT -i lo -j ACCEPT[9]
```

```
iptables -A OUTPUT -o lo -j ACCEPT[9]
```



```

root@bjoshi:/home/bjoshi# iptables -F
root@bjoshi:/home/bjoshi# iptables -A OUTPUT -p icmp --icmp-type echo-request -j DROP
root@bjoshi:/home/bjoshi# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
DROP      icmp -- 0.0.0.0/0             0.0.0.0/0             icmp: 8
root@bjoshi:/home/bjoshi# ping 10.11.15.62
PING 10.11.15.62 (10.11.15.62) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
^C
--- 10.11.15.62 ping statistics ---
7 packets transmitted, 0 received, 100% packet loss, time 6046ms

root@bjoshi:/home/bjoshi# iptables -F
root@bjoshi:/home/bjoshi# ping 10.11.15.62
PING 10.11.15.62 (10.11.15.62) 56(84) bytes of data.
64 bytes from 10.11.15.62: icmp_seq=1 ttl=64 time=0.195 ms
64 bytes from 10.11.15.62: icmp_seq=2 ttl=64 time=0.145 ms
64 bytes from 10.11.15.62: icmp_seq=3 ttl=64 time=0.174 ms
64 bytes from 10.11.15.62: icmp_seq=4 ttl=64 time=0.198 ms
^C
--- 10.11.15.62 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.145/0.178/0.198/0.021 ms
root@bjoshi:/home/bjoshi# █

```

Figure 5.2: iptable rule

Above Figure 5.2 show the IPTable rule for blocking ICMP echo request, after applying this rule in OUTPUT chain the connection request request ping with ip 10.11.15.62 is blocked.

```

bjoshi:/home/bjoshi
bjoshi@bjoshi:~$ sudo su
[sudo] password for bjoshi:
root@bjoshi:/home/bjoshi# ping 10.11.15.62
PING 10.11.15.62 (10.11.15.62) 56(84) bytes of data.
64 bytes from 10.11.15.62: icmp_seq=1 ttl=64 time=0.397 ms
64 bytes from 10.11.15.62: icmp_seq=2 ttl=64 time=0.192 ms
64 bytes from 10.11.15.62: icmp_seq=3 ttl=64 time=0.242 ms
64 bytes from 10.11.15.62: icmp_seq=4 ttl=64 time=0.245 ms
64 bytes from 10.11.15.62: icmp_seq=5 ttl=64 time=0.249 ms
64 bytes from 10.11.15.62: icmp_seq=6 ttl=64 time=0.116 ms
64 bytes from 10.11.15.62: icmp_seq=7 ttl=64 time=0.212 ms
64 bytes from 10.11.15.62: icmp_seq=8 ttl=64 time=0.282 ms
^C
--- 10.11.15.62 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 6999ms
rtt min/avg/max/mdev = 0.116/0.241/0.397/0.077 ms
root@bjoshi:/home/bjoshi# iptables -A INPUT -s 10.11.15.62 -j DROP
root@bjoshi:/home/bjoshi# ping 10.11.15.62
PING 10.11.15.62 (10.11.15.62) 56(84) bytes of data.
^C
--- 10.11.15.62 ping statistics ---
12 packets transmitted, 0 received, 100% packet loss, time 11087ms

root@bjoshi:/home/bjoshi# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
DROP      all  --  10.11.15.62            0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
root@bjoshi:/home/bjoshi# iptables -F
root@bjoshi:/home/bjoshi# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
root@bjoshi:/home/bjoshi# iptables -A INPUT -s 10.11.15.62 -j ACCEPT
root@bjoshi:/home/bjoshi# ping 10.11.15.62
PING 10.11.15.62 (10.11.15.62) 56(84) bytes of data.
64 bytes from 10.11.15.62: icmp_seq=1 ttl=64 time=0.211 ms
64 bytes from 10.11.15.62: icmp_seq=2 ttl=64 time=0.219 ms
64 bytes from 10.11.15.62: icmp_seq=3 ttl=64 time=0.238 ms
64 bytes from 10.11.15.62: icmp_seq=4 ttl=64 time=0.223 ms
64 bytes from 10.11.15.62: icmp_seq=5 ttl=64 time=0.225 ms
^C
--- 10.11.15.62 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.211/0.223/0.238/0.012 ms
root@bjoshi:/home/bjoshi#

```

Figure 5.3: iptable rule

Above figure 5.3 show that using IPTable rule, block the IPaddress from our computer by dropping the IPaddress in IPTable rule from INPUT chain.

```

num target      prot opt source          destination
Chain OUTPUT (policy ACCEPT)
num target      prot opt source          destination
1 ACCEPT        all  -- anywhere      anywhere
root@bjoshi:/home/bjoshi# host -t a www.facebook.com
www.facebook.com is an alias for star-mini.c10r.facebook.com.
star-mini.c10r.facebook.com has address 31.13.78.35
root@bjoshi:/home/bjoshi# iptables -A OUTPUT -p tcp -d ^Cj DROP
root@bjoshi:/home/bjoshi# iptables -A OUTPUT -p tcp -d 31.13.78.35 -j DROP
root@bjoshi:/home/bjoshi# iptables -L --line-number
Chain INPUT (policy ACCEPT)
num target      prot opt source          destination
1 ACCEPT        all  -- anywhere      anywhere

Chain FORWARD (policy ACCEPT)
num target      prot opt source          destination

Chain OUTPUT (policy ACCEPT)
num target      prot opt source          destination
1 ACCEPT        all  -- anywhere      anywhere
2 DROP          tcp  -- anywhere      edge-star-mini-shv-01-sit4.facebook.com
root@bjoshi:/home/bjoshi#/sbin/iptables-save
bash: /sbin/iptables-save: No such file or directory
root@bjoshi:/home/bjoshi# iptables -A INPUT -p tcp -m tcp -d www.facebook.com -j DROP
root@bjoshi:/home/bjoshi# iptables -L --line-number
Chain INPUT (policy ACCEPT)
num target      prot opt source          destination
1 ACCEPT        all  -- anywhere      anywhere
2 DROP          tcp  -- anywhere      edge-star-mini-shv-01-sit4.facebook.com tcp

Chain FORWARD (policy ACCEPT)
num target      prot opt source          destination

Chain OUTPUT (policy ACCEPT)
num target      prot opt source          destination
1 ACCEPT        all  -- anywhere      anywhere
2 DROP          tcp  -- anywhere      edge-star-mini-shv-01-sit4.facebook.com
root@bjoshi:/home/bjoshi# ^C
root@bjoshi:/home/bjoshi# -A INPUT -p tcp -m tcp -d www.youtube.com -j DROP
-A: command not found
root@bjoshi:/home/bjoshi# iptables -A INPUT -p tcp -m tcp -d www.youtube.com -j DROP
root@bjoshi:/home/bjoshi# iptables -L --line-number
Chain INPUT (policy ACCEPT)
num target      prot opt source          destination
1 ACCEPT        all  -- anywhere      anywhere
2 DROP          tcp  -- anywhere      edge-star-mini-shv-01-sit4.facebook.com tcp
3 DROP          tcp  -- anywhere      maa03s20-in-f46.1e100.net tcp

Chain FORWARD (policy ACCEPT)
num target      prot opt source          destination

Chain OUTPUT (policy ACCEPT)
num target      prot opt source          destination
1 ACCEPT        all  -- anywhere      anywhere
2 DROP          tcp  -- anywhere      edge-star-mini-shv-01-sit4.facebook.com
root@bjoshi:/home/bjoshi# host -t a www.youtube.com
www.youtube.com is an alias for youtube-ut.l.google.com.

```

Figure 5.4: iptable rule

Above figure 5.4 show the IPTable rule for blocking the site like facebook or youtube from INPUT chain and OUTPUT chain.

Chapter 6

Analysis of Web user Interface

6.1 Webmin

Webmin prevent the use of doing configuration task using command line interface in linux, it is a web user interface. Webmin is user interface provide various feature such as setting up account, HostAPD security, apache server, DNS for mapping domain name to IP address, DHCP, time management, log file, file sharing. Webmin is used for adding firewall rule, set time, different theme's are there in webmin,we can setup system log file, Edit the configuration of SSH server, DHCP server, HostAPD, apache host server.

Other module require to configure other server and its services, but webmin module configure its own webmin.

It lets you do things like change the port and Webmin uses, limit the client addresses that can connect, change the theme and language that the user interface uses and install new modules[12].

Webmin has third party module and 113 standard module, third party module is added by us and can be used whenever required for adding function. Webmin modules are written in perl scripting language. Modules are collection of server, system, hardware, webmin, networking, cluster and unused module[12].

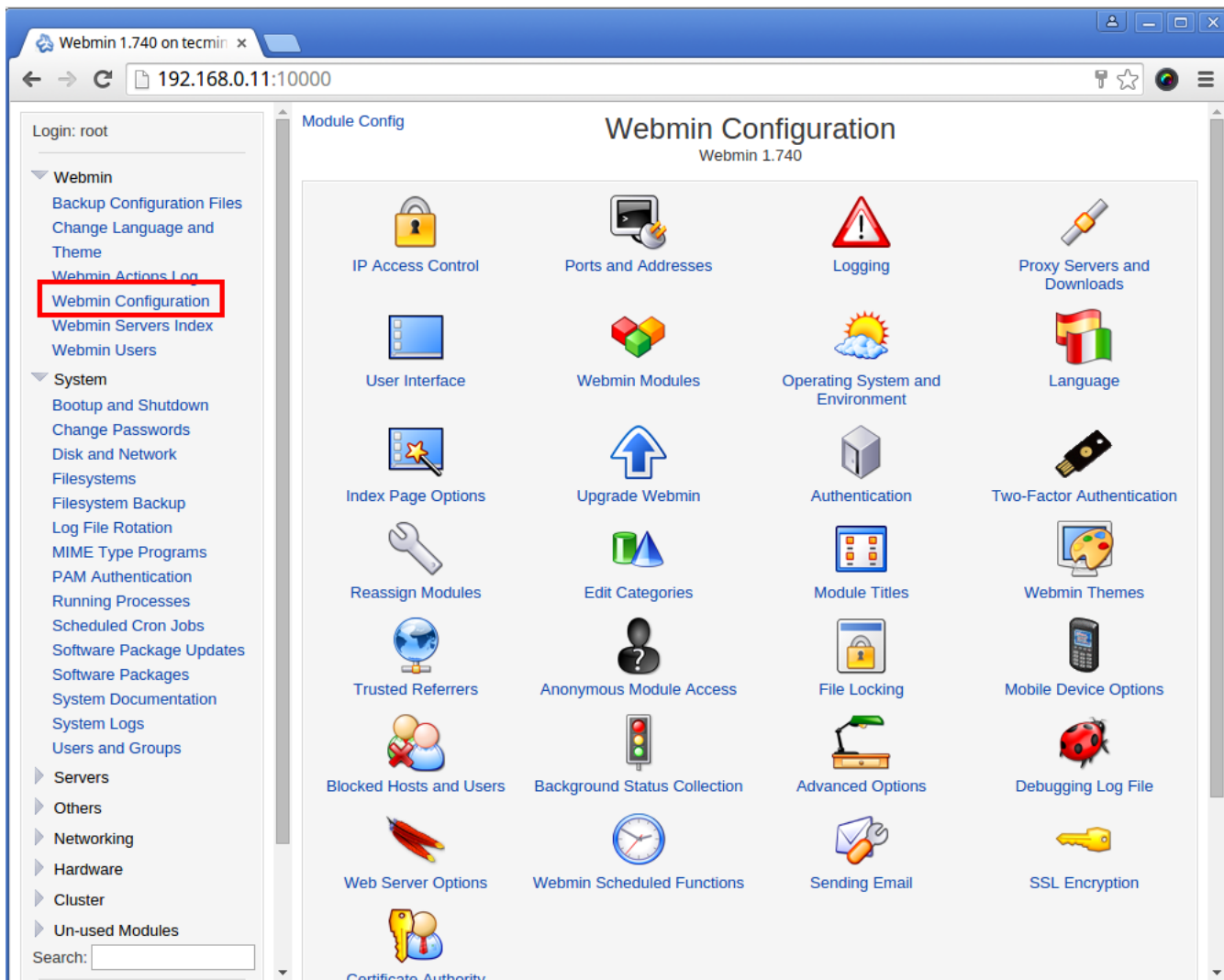


Figure 6.1: Webmin configuration [13]

Above figure 6.1 show the webmin configuration for webmin.

IP access control is for controlling the number of IP address which can access the webmin server , we can make a list of number of IP address which can access the webmin server and other will be blocked by the user.

Webmin is allowed or work on port number 10000, so that any client can access this port, but to this threat of malicious activity are increased , so we can change the port number in firewall rule like 443 or 80.

Chapter 7

Wifi Provider for thing Devices

To create environment for providing wifi to n devices in smart home we need to analyze different modules like hostapd and isc-dhcp-server and develop there functionality. We have dependencies of intel nuc device with linux(ubuntu) system and modules such as hostapd, wpa authenticator, isc-dhcp-server, iptables.

7.1 HostAPD

HostAPD module use to create a software access point which is to be access by different client in smart home by providing broadcast ssid and authenticate various client which are connected to these software access point. I create a software access point in linux system and provide wifi to various devices in smart home which are connected to these access point.

Hostapd is an IEEE 802.11 AP and IEEE 802.1X/WPA/WPA2/EAP/RADIUS Authenticator. To communicate with a kernel driver hostapd has to use some interface. All new cfg80211 (and mac80211) based drivers that implement AP functionality are supported using nl80211 interface.

Installing Hostapd in Linux system

```
sudo apt-get update && sudo apt-get install hostapd
```

7.2 Hostapd Configuration

Hostapd Configuration File

\$etc/hostapd/hostapd.conf file is the main file which is to be analyse and implement for creating software access point.

Write a hostapd.conf file with following content specified in below figure 7.2

Start it with the following command:

```
$sudo hostapd hostapd.conf
```

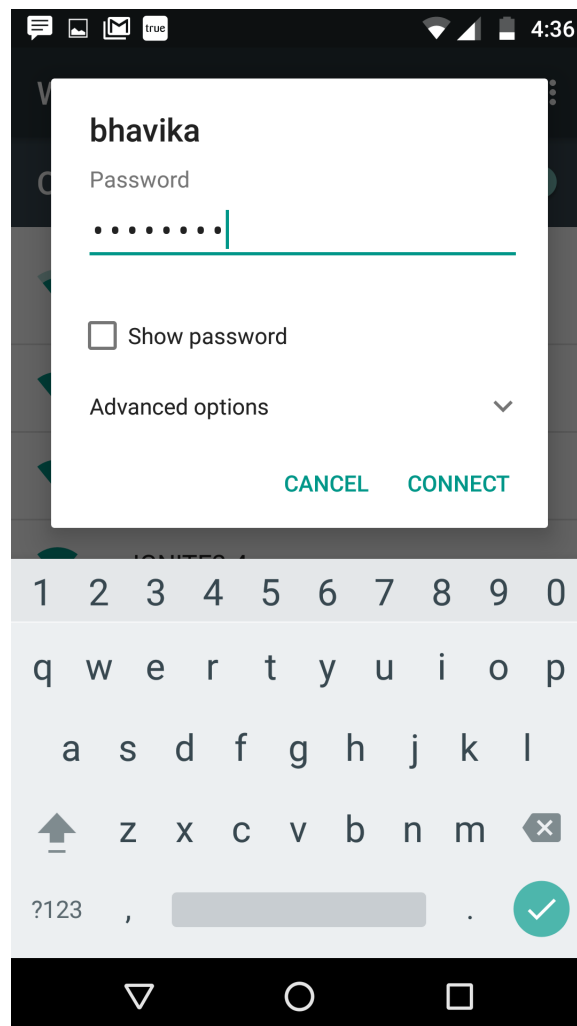


Figure 7.1: Mobile connected with enable Access Point

You have to check your software access point is working or not using any wifi enable device like mobile phone or laptop, still you are not able to connect to it at present. Only you can check broadcast ssid and authenticate to this software access point. While your hostapd is working fine you need to set up dhcp module for providing IP address to different client which are connected to your access point. Above figure 7.1 show the mobile connected with enable access point

```
1 #sets the wifi interface to use, is wlan0 in most cases
2 interface=wlan0
3 #sets the ssid of the virtual wifi access point
4 ssid=bhavika
5 #sets the mode of wifi, depends upon the devices you will be using. It can be a,b,g,n
6 hw_mode=g
7 #sets the channel for your wifi
8 channel=1
9 #macaddr_acl sets options for mac address filtering. 0 means "accept unless in deny 1
10 macaddr_acl=0
11 #setting ignore_broadcast_ssid to 1 will disable the broadcasting of ssid
12 ignore_broadcast_ssid=0
13 #Sets authentication algorithm
14 #1 - only open system authentication
15 #2 - both open system authentication and shared key authentication
16 auth_algs=1
17
18 #####Sets WPA and WPA2 authentication#####
19 #wpa option sets which wpa implementation to use
20 #1 - wpa only
21 #2 - wpa2 only
22 #3 - both
23 wpa=3
24 #sets wpa passphrase required by the clients to authenticate themselves on the network
25 wpa_passphrase=12345678
26 #sets wpa key management
27 wpa_key_mgmt=WPA-PSK
28 #sets encryption used by WPA
29 wpa_pairwise=TKIP
30 #sets encryption used by WPA2
31 rsn_pairwise=CCMP|
```

Figure 7.2: Hostapd configuration file

Next step is to enable IPv4 forwarding:

/etc/sysctl.conf file uncomment the

```
#net.ipv4.ip_forward=1
```

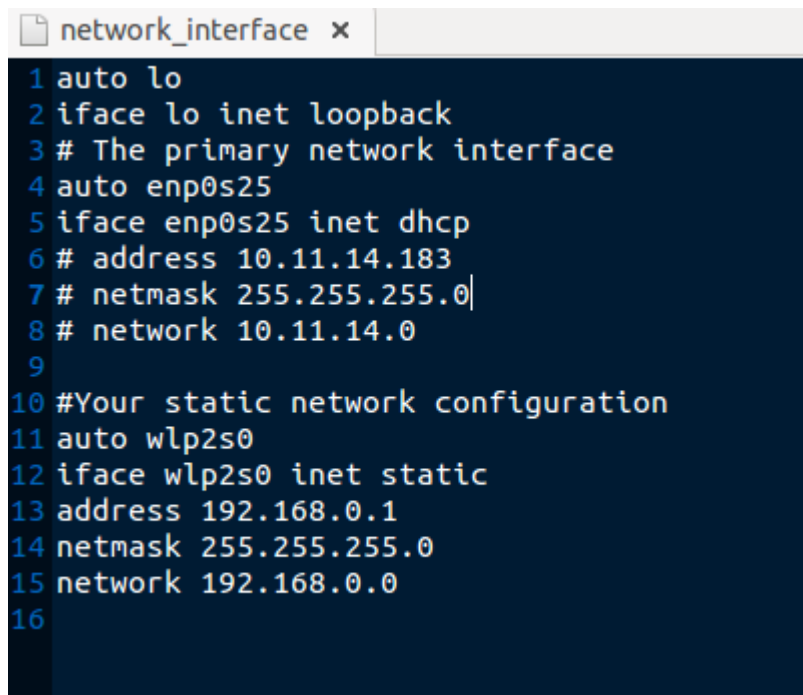
7.3 Network Interface File

Before we configure dhcpd config file first we need to make changes in our network interface file.

The interface file found in etc directory network folder:

/etc/network/interfaces file

We have to make our eth0 ethernet link interface dynamic and wlan0 wireless interface static in our network interface file.



```
network_interface x
1 auto lo
2 iface lo inet loopback
3 # The primary network interface
4 auto enp0s25
5 iface enp0s25 inet dhcp
6 # address 10.11.14.183
7 # netmask 255.255.255.0
8 # network 10.11.14.0
9
10 #Your static network configuration
11 auto wlp2s0
12 iface wlp2s0 inet static
13 address 192.168.0.1
14 netmask 255.255.255.0
15 network 192.168.0.0
16
```

Figure 7.3: Network Interface file

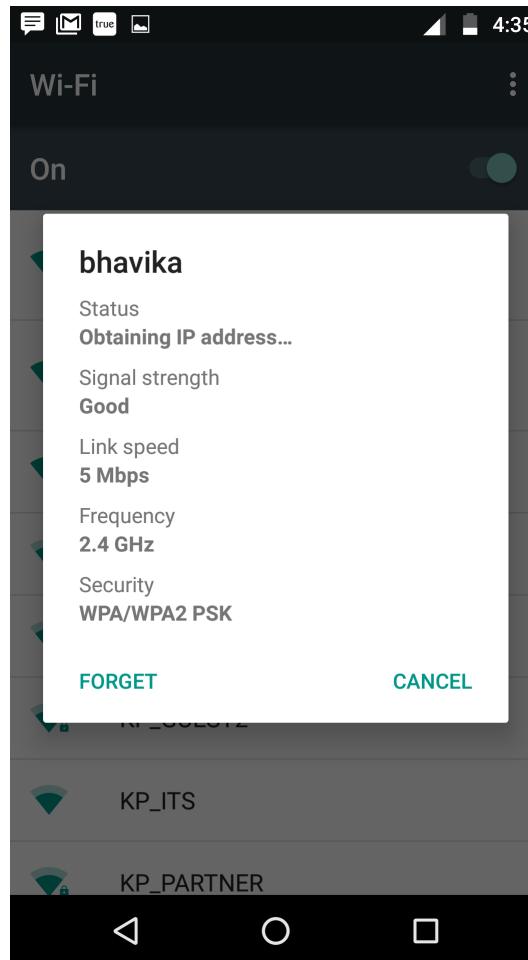


Figure 7.4: Mobile device connected to access point

7.4 DHCP server

While our `hostapd` is working fine, we need to set up `dhcp` server to provide ip address to all client which are connected to our access point in our smart home. Dynamic Host Configuration Protocol (DHCP) is a client/server protocol that automatically provides an Internet Protocol (IP) host with its IP address and other related configuration information such as the subnet mask and default gateway[16].

Installing `isc-dhcp-server` in Linux system:

```
$sudo apt-get update && sudo apt-get install isc-dhcp-server
```

7.5 DHCP Configuration

\$etc/dhcp/dhcpd.conf file is the main file which is to be analyse and implement for providing ip address to all client which are connected to access point.

Steps need to follow in DHCP configuration file are shown in figure 7.5:

```
1 ddns-update-style none;
2 log-facility local7;
3 subnet 192.168.0.0 netmask 255.255.255.0 {
4     range 192.168.0.100 192.168.0.254;
5     option broadcast-address 192.168.0.255;
6     option routers 192.168.0.1;
7     option subnet-mask 255.255.255.0;
8     default-lease-time 600;
9     max-lease-time 7200;
10    option domain-name "local-network";
11    option domain-name-servers 8.8.8.8, 8.8.4.4;
12 }
13
```

Figure 7.5: DHCP configuration file

To check status of dhcp and hostapd command used are:

```
$sudo service isc-dhcp-server status
```

```
$sudo service hostapd status
```

To restart dhcp and hostapd command used are:

```
$sudo service isc-dhcp-server restart
```

```
$sudo service hostapd restart
```

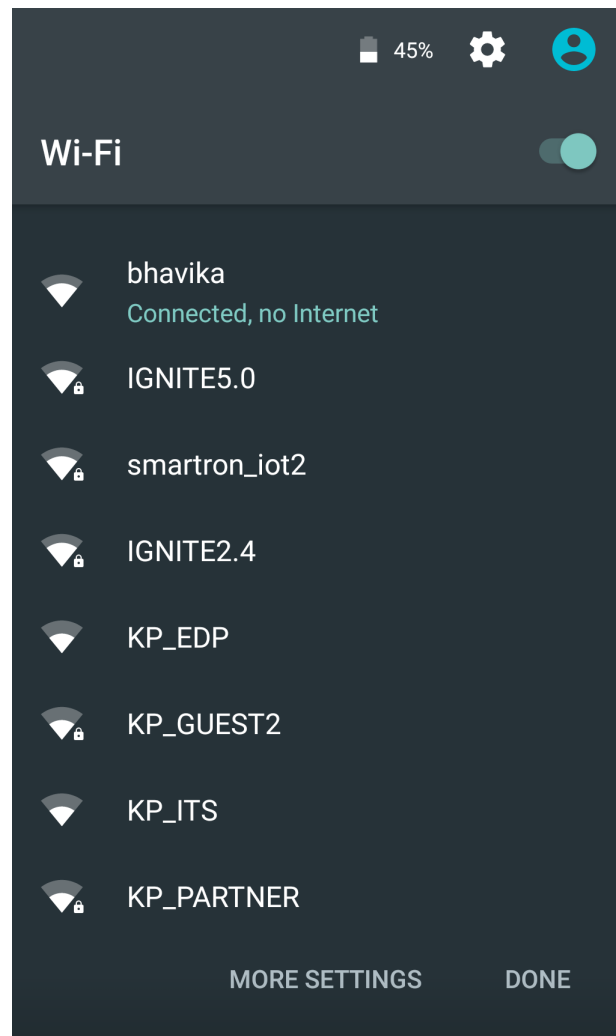


Figure 7.6: Mobile device connected to access point

7.6 Network Address Translation

To share internet in one network interface with the clients connected through hostapd we need to enable network address translation.

Steps followed to enable NAT: Configure iptables for NAT translation so that packets can be correctly routed through the gateway.

```
$sudo iptables -F FORWARD
```

```
$sudo iptables -A FORWARD -o enp0s25 -i wlp2s0 -s 192.168.0.0/24 -m conntrack
```

```
-ctstate NEW -j ACCEPT
```

```
$sudo iptables -A FORWARD -m conntrack -ctstate ESTABLISHED,RELATED -j  
ACCEPT
```

```
$sudo iptables -t nat -F POSTROUTING
```

```
$sudo iptables -t nat -A POSTROUTING -o enp0s25 -j MASQUERADE
```

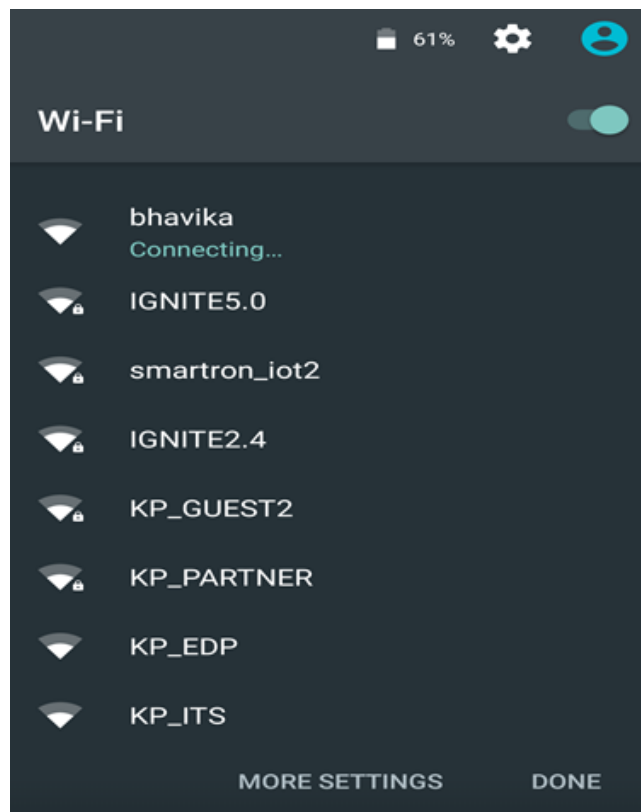


Figure 7.7: Mobile device connected to access point

To provide internet to clients connected through hostapd first we need to disable our network manager to disable our device connected to wifi.

```
$ sudo systemctl stop NetworkManager.service
```

```
$ sudo systemctl disable NetworkManager.service
```

I have included all the steps to configure wlp2s0 interface, enable NAT, start DHCP server and hostapd in the BASH script below

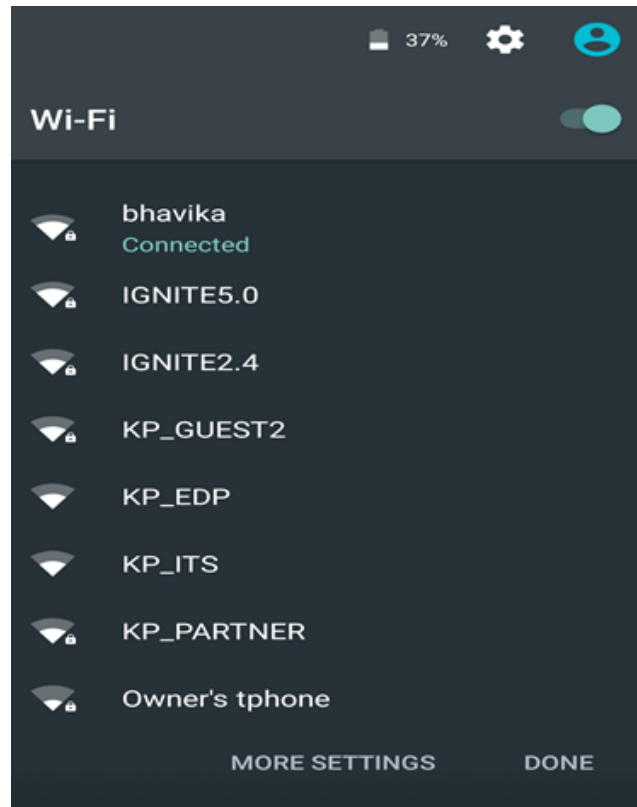


Figure 7.8: Mobile device connected to access point

In this script we call all the file of hostapd, dhcpd, enable ipv4 forwarding, enable NAT and configure wlp2s0 interface.

Pseudo code for boot time script to start our intel NUC board as a router-

We call this script at boot time so that intel nuc board work as router whenever we start our PC.

```
#!/bin/sh
#sudo nmcli radio wifi off
#sudo rfkill list
#sudo rfkill unblock wlan
#sudo rfkill list
sudo systemctl stop NetworkManager.service
sudo systemctl disable NetworkManager.service
sudo ifconfig wlp2s0 192.168.0.1/24 up
sleep 2
sudo ifconfig
#sudo service isc-dhcp-server restart
#sudo service isc-dhcp-server status
#sudo service hostapd restart
#sudo service hostapd status
#Doesn't try to run dhcpd when already running
if [ "$(ps -e | grep dhcpd)" == "" ]; then
dhcpd wlp2s0 &
fi

sysctl -w net.ipv4.ip_forward=1

#start hostapd
hostapd /etc/hostapd/hostapd.conf 1>/dev/null
#killall dhcpd
```

Figure 7.9: Boot time script file

Above figure 7.9 show the boot time script, We call this script during boot time, It call each module Hostapd, dhcp, network interface file during boot time and enable our wifi so that thing device's in T-home can connect to these wifi and start communicating.

Chapter 8

thome RESTful API Development Standard

Back end API for Router are written using node.js script with module support of wireless-tools and Express module. NodeJs is a platform built on chromes V8 engine for easily building fast and scalable network apps and it uses an event driven,non-blocking I/O model that makes it lightweight & efficient. Node is non blocking input output function which use callback function to return result.

8.1 Technology

Nodejs:

NodeJs is a platform built on chromes V8 engine for easily building fast and scalable network apps and It uses an event driven, non-blocking I/O model that makes it lightweight & efficient.

Step to install nodejs in ubuntu-

```
sudo apt-get install nodejs
```

```
sudo apt-get update
```


NPM:

Command to install npm in ubuntu

```
$ sudo apt-get install npm
```

NPM is a CLI tool program to manage NodeJs libs,

Simple to install node.js module using NPM library-

```
npm install <module name >
```

Example for installing module name using npm:

```
npm install express
```

```
npm install wireless-tools
```

8.2 Development Environment

Atom :

Atom is an open source IDE with support for plugins and embedded git control. Atom work on different operating system Linux, windows used for auto complete the task on these editor.

jshint jhint eslint plugins :

JavaScript code quality analysis tools (add as plugins for ATOM)

Running:

Curl - CLI based tool

Postman - Chrome plugin (GUI) for checking output of nodejs program

8.3 Documentation

apiDoc creates documents automatically from API annotations from your source code, you have to follow the apidocs annotation standards when writing the code, Include the supported list of error codes in annotations

8.4 Testing

Every API should be tested by the developer before pushing it to git. A folder named Test should be available in every project source, which should contain basic test cases of implemented API.

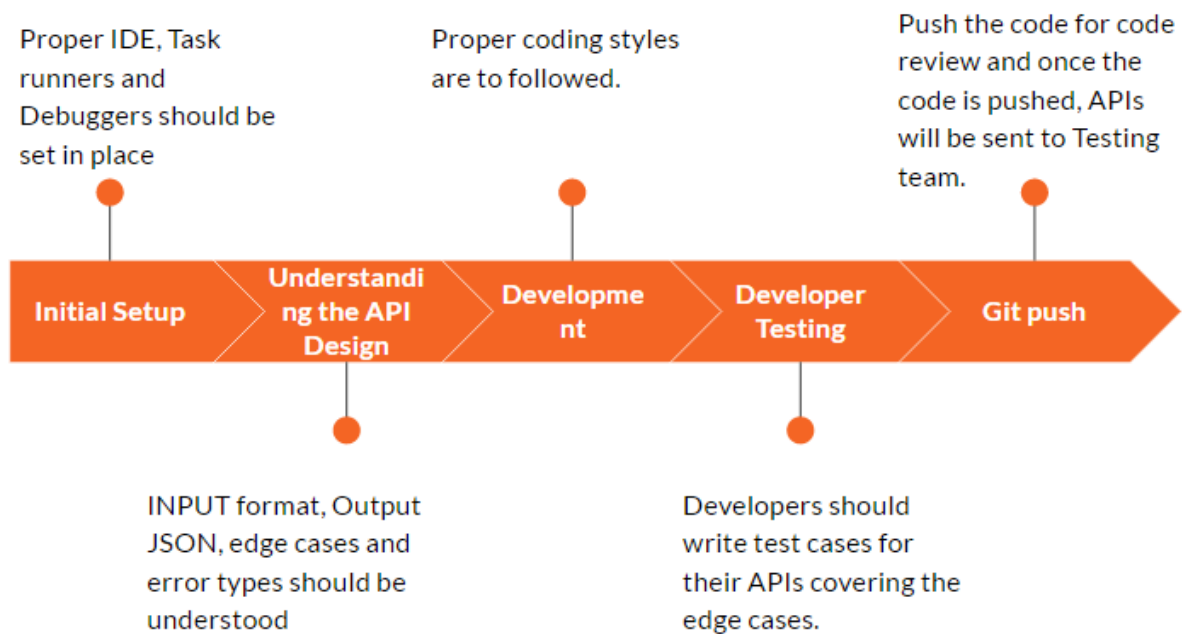


Figure 8.1: Router Restful API Development Standard

Above figure 8.1 show the process of API standards for publishing.

Wireless-tools: Wireless-tools is npm module which contain all the function related to router dependencies like hostapd, dhcp, iw, wpa_supplicant, ifconfig which is used to develop backend api for web user interface of router. It is the most secure and less issue are there in these module.

Wireless-tools/HostAPD: It is used to create software access point on particular wireless network interface. It have functionality to enable all the variable in hostapd config file required to create access point and run hostapd in background. It will also

provide functionality to disable the hostapd which is created on particular wireless interface.

Wireless-tools/ifconfig: It is used to get the detail of each network interface in your system like wireless interface, ethernet interface, local host interface details, and provide information related to `ipv4_address`, `broadcast_address`, `subnet_mask`, `ipv6_address`, `interface`, `mac_address`, `link` etc. It also provide functionality to update any details of particular interface related tp `ipv4_address`, its `subnet_mask`, `broadcast_address`. It also provide functionality to disable any interface on your system.

HTTP/ifconfig?interface=eth0 status 200 ok

```
[
{
"interface": "eth0",
"link": "ethernet",
"address": "58:91:cf:c2:e7:aa",
"ipv4_address": "10.11.15.169",
"ipv4_broadcast": "10.11.255.255",
"ipv4_subnet_mask": "255.255.0.0",
"broadcast": true,
"multicast": true
}
]
```

Wireless-tools/iwconfig: It is used to get the details of all wireless interface enable on your system. It provide information related to wireless interface such as to which access point it is connected, its ssid, its operating mode, frequency band, wifi standard. Also it provide information related to each particular wireless interface if we want to render particular wireless interface related information.

HTTP/iwconfig?interface=iwconfig status 200 ok

```
[
{
```

```
"interface": "wlan0",
"access_point": "ec:8c:a2:18:52:8c",
"frequency": 5.2,
"ieee": "802.11abgn",
"mode": "managed",
"quality": 50,
"signal": -60,
"ssid": "Smartron"
}
]
```

Wireless-tools/iwlist: It is used to know which access point are enable at particular region related to wireless interface card, and parse all the access point status. API is based on query the status of network interface of device using get function and it return data in JSON format.

Parses the status of all wireless interface file.

```
HTTP/iwlist?interface=wlan0 status 200 ok
[
{
"address": "ec:8c:a2:58:55:a8",
"channel": 1,
"frequency": 2.412,
"mode": "master",
"quality": 63,
"signal": -47,
"ssid": "SmartronGuest",
"security": "open"
} ]
```

Chapter 9

Conclusion

Developed functionality for router modules, able to fulfill various requirements such as providing internet connectivity to end things, home users devices in smart home. This would help in controlling all things devices in smart home. For security developed firewall rules for user, many companies have their own firewall rules, For restricting any illegal activity in smart home router developed a firewall rule.

For backend process , For web user interface of router developed backend API related to ifconfig, iwconfig, iwlist, iwscan, wpa_cli, wpa_supplicant, hostapd, dhcp module. By providing web user interface user can easily configure router module according to their requirements.

By using latest technology and tools and adding advance features in our router we give end user amazing experience with easiness in using this router in smart home .

Chapter 10

Future Scope

The current feature in router are not advance enough for smarthome, Adding new feature for better connectivity and security making router more flexible and advance for end-user. For security purpose,latest anti-virus can be added.

Currently the backend API developed for router 0.1.0 version, after which there are many changes that have been done in the user interface and that result into release of router 0.2.0 version.

Bibliography

- [1] "Securifi Almond", [Online], Website, December 2016,
[http : //www.tomsguide.com/us/securifi – almond – router,review – 2823.html](http://www.tomsguide.com/us/securifi-almond-router,review-2823.html)

- [2] "Chime", [Online], Website, December 2016,
[http : //www.chimewifi.com/](http://www.chimewifi.com/)

- [3] "Torch feature", [Online], Website, December 2016,
[https : //mytorch.com/pages/features](https://mytorch.com/pages/features)

- [4] "Eero", [Online], Website, December 2016,
[https : //eero.com/](https://eero.com/)

- [5] "Banana Pi R1 Overview", [Online], Website, December 2016,
[http : //www.banana – pi.org/r1.html](http://www.banana-pi.org/r1.html)

- [6] "Building OpenWRT trunk on a BPI-R1", [Online], Website, December 2016,
[http : //wiki.geiges.net/doku.php?id = openwrt_pi – r1](http://wiki.geiges.net/doku.php?id=openwrt_pi-r1)

- [7] "IPtables", [Online], Website, December 2016,
[https : //access.redhat.com/documentation/en – US/RedHatEnterpriseLinux/3/html/ReferenceGuide/s1 – iptables – options.html](https://access.redhat.com/documentation/en-US/RedHatEnterpriseLinux/3/html/ReferenceGuide/s1-iptables-options.html)

- [8] "Linux Firewall Tutorial: IPTables Tables, Chains, Rules Fundamentals", [Online], Website, December 2016,
[http : //www.thegeekstuff.com/2011/01/iptables - fundamentals](http://www.thegeekstuff.com/2011/01/iptables-fundamentals)
- [9] " How To Set Up a Firewall Using Iptables on Ubuntu 14.04 ", [Online], Website, December 2016,
[https : //www.digitalocean.com/community/tutorials/how - to - set - up - a - firewall - using - iptables - on - ubuntu - 14 - 04](https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-using-iptables-on-ubuntu-14-04)
- [10] "Sysresccd-Networking-EN-Iptables-and-netfilter-load-balancing-using-connmack", [Online], Website, December 2016,
[https : //www.system - rescue - cd.org/Sysresccd - Networking - EN - Iptables - and - netfilter - load - balancing - using - connmack](https://www.system-rescue-cd.org/Sysresccd-Networking-EN-Iptables-and-netfilter-load-balancing-using-connmack)
- [11] "Router", [Online], Website, December 2016,
[https : //en.wikipedia.org/wiki/Router\(computing\)](https://en.wikipedia.org/wiki/Router(computing))
- [12] "Webmin Configuration", [Online], Website, December 2016,
[http : //doxfer.webmin.com/Webmin/Webmin_Configuration](http://doxfer.webmin.com/Webmin/Webmin_Configuration)
- [13] "Webmin based system", [Online], Website, December 2016,
[http : //www.tecmint.com/install - webmin - in - centos - rhel - fedora - ubuntu - debian/](http://www.tecmint.com/install-webmin-in-centos-rhel-fedora-ubuntu-debian/)
- [14] "Banana Pi R1", [Online], Website, December 2016,
[http : //www.banana - pi.org/images/bpi - images/R1/r3.jpg](http://www.banana-pi.org/images/bpi-images/R1/r3.jpg)
- [15] "Net gear QOS", [Online], Website, December 2016,
[http : //support1.gearguy.com/useruploads/images/2\(18\).png](http://support1.gearguy.com/useruploads/images/2(18).png)
- [16] "DHCP", [Online], Website, December 2016,
[https : //technet.microsoft.com/en - us/library/dd145320\(v = ws.10\).aspx](https://technet.microsoft.com/en-us/library/dd145320(v=ws.10).aspx)

- [17] "HostAPD", [Online], Website, December 2016,
<https://wiki.gentoo.org/wiki/Hostapd>
- [18] "OpenWRT", [Online], Website, December 2016,
<https://openwrt.org/>
- [19] "Operating Mode", [Online], Website, December 2016,
<https://wireless.wiki.kernel.org/en/users/Documentation/modes>
- [20] "wpa", [Online], Website, December 2016,
https://wiki.archlinux.org/index.php/WPA_supplicant
- [21] "wpacli", [Online], Website, December 2016,
https://linux.die.net/man/8/wpa_cli