

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**----------**

**LUẬN VĂN TỐT NGHIỆP**



**Đề tài**

**CÂN BẰNG TẢI HỆ THỐNG WI-FI BẰNG BỘ ĐIỀU KHIỂN TẬP TRUNG**

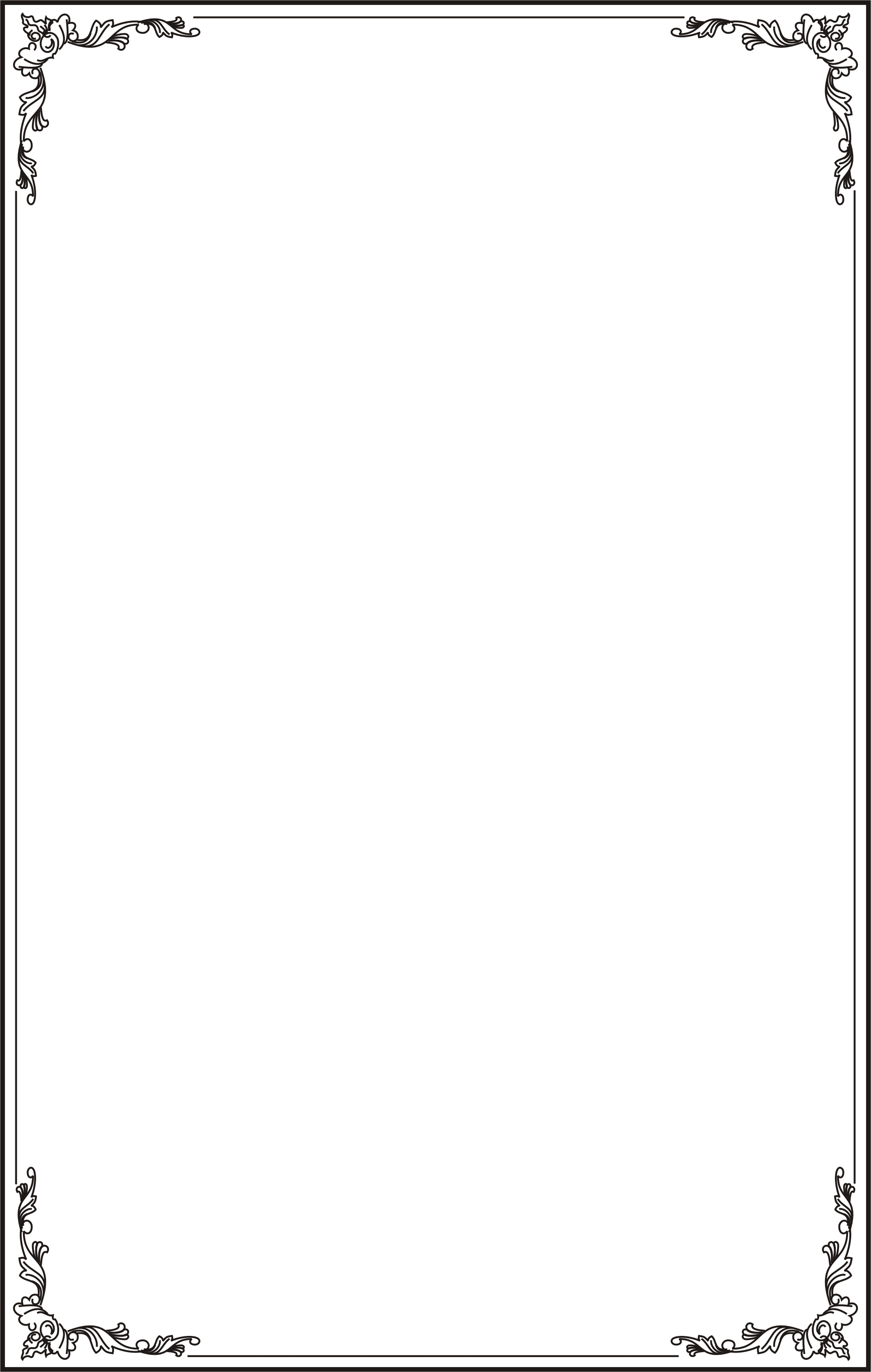
***Sinh viên thực hiện***

***Phạm Quốc Khải - B1401149***

***Giáo viên hướng dẫn***

***• TS. Thái Minh Tuấn***

**Cần Thơ, 12/2018**



**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**----------**

**LUẬN VĂN TỐT NGHIỆP**



**Đề tài**

**CÂN BẰNG TẢI HỆ THỐNG WI-FI**

**BẰNG BỘ ĐIỀU KHIỂN TẬP TRUNG**

***Sinh viên thực hiện***

***Phạm Quốc Khải - B1401149***

***Giáo viên hướng dẫn***

***• TS. Thái Minh Tuấn***

**Cần Thơ, 12/2018**

# NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

……….Ngày…...tháng.…..năm 2018

Giáo viên hướng dẫn

*(Kí tên)*

# LỜI CẢM ƠN

Để hoàn thành tốt luận văn này, tôi đã nhận được sự giúp đỡ, hỗ trợ và động viên của rất nhiều cá nhân cũng như tập thể. Nhân đây, chúng tôi xin tỏ lòng biết ơn đến gia đình đã quan tâm, ủng hộ về vật chất lẫn tinh thần trong quá trình chúng tôi học tập và thực hiện luận văn.

Xin chân thành biết ơn sâu sắc đến thầy Thái Minh Tuấn đã rất tận tình hướng dẫn, góp ý, truyền đạt những kiến thức và kinh nghiệm quý báu để chúng tôi hoàn thành tốt luận văn này.

Chân thành cảm ơn quý thầy cô bộ môn Công nghệ thông tin - Đại học Cần Thơ đã tận tình chỉ bảo, giúp đỡ trong thời gian chúng tôi thực hiện luận văn. Cuối cùng chúng chúng tôi tin cảm ơn bạn bè, anh chị khoa Công nghệ thông tin và Truyền thông đã cho chúng chúng tôi ý tưởng, hỗ trợ, động viên chúng tôi trong quá trình thực hiện luận văn. Mặc dù cố gắng hoàn thiện luận văn này nhưng do kiến thức còn hạn chế nên không thể tránh những sai sót, vì thế chúng chúng tôi mong nhận được những góp ý của thầy cô và các bạn.

**PHẠM QUỐC KHẢI**

# 

# MỤC LỤC

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN i](#_Toc19537)

[LỜI CẢM ƠN ii](#_Toc13920)

[MỤC LỤC iv](#_Toc29785)

[DANH SÁCH CÁC TỪ VIẾT TẮT vi](#_Toc18030)

[DANH SÁCH HÌNH vii](#_Toc27361)

[DANH SÁCH BẢNG ix](#_Toc30317)

[TÓM TẮT x](#_Toc10668)

[ABSTRACT xi](#_Toc5435)

[CHƯƠNG 1: TỔNG QUAN 1](#_Toc9608)

[1.1 ĐỘNG CƠ NGHIÊN CỨU 1](#_Toc12833)

[1.2 MỤC TIÊU 1](#_Toc28807)

[1.3 NHỮNG NGHIÊN CỨU LIÊN QUAN 1](#_Toc29362)

[1.4 PHƯƠNG PHÁP NGHIÊN CỨU 2](#_Toc13903)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 3](#_Toc18057)

[2.1 MẠNG WI-FI 3](#_Toc6234)

[2.1.1 Mạng Wi-Fi là gì? 3](#_Toc8289)

[2.1.2 Nguyên tắc hoạt động của mạng Wi-Fi 4](#_Toc16652)

[2.1.3 Roaming trong mạng Wi-Fi 4](#_Toc24890)

[2.2 FIRMWARE OPENWRT 6](#_Toc13523)

[2.2.1 Firmware OpenWrt là gì? 6](#_Toc11661)

[2.2.2 Những tính năng của OpenWrt 6](#_Toc30974)

[2.3 TFTP 10](#_Toc1346)

[2.4 TIẾN TRÌNH KHỞI ĐỘNG ROUTER 11](#_Toc9820)

[2.5 GIỚI THIỆU CÔNG NGHỆ SDN 11](#_Toc31820)

[2.5.1 SDN là gì? 11](#_Toc28148)

[2.5.2 Giao thức OpenFlow 12](#_Toc8458)

[2.5.3 OpenVSwitch 12](#_Toc2123)

[2.6 PYTHON 13](#_Toc18153)

[2.7 RYU FRAMEWORK 13](#_Toc3098)

[2.8 CÔNG NGHỆ WEB 13](#_Toc24281)

[CHƯƠNG 3: NỘI DUNG VÀ KẾT QUẢ NGHIÊN CỨU 18](#_Toc20313)

[3.1 NỘI DUNG NGHIÊN CỨU 18](#_Toc5259)

[3.1.1 Lựa chọn thiết bị 19](#_Toc19986)

[3.1.2 Cài đặt firmware cho router 20](#_Toc24219)

[3.1.3 Lựa chọn hình trạng mạng 21](#_Toc11350)

[3.1.4 Xây dựng bộ điều khiển 23](#_Toc24850)

[3.1.5 Thiết kế giải thuật cân bằng tải 24](#_Toc21830)

[3.1.6 Xây dựng website theo dõi hệ thống 25](#_Toc15585)

[3.2 KẾT QUẢ NGHIÊN CỨU 29](#_Toc1819)

[3.2.1 Hệ thống topology openwrt 29](#_Toc31810)

[3.2.2 Controller điều khiển hệ thống 29](#_Toc23677)

[3.2.3 Website quản trị 31](#_Toc2856)

[CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 33](#_Toc9663)

[4.1 KẾT LUẬN 33](#_Toc25635)

[4.2 HƯỚNG PHÁT TRIỂN 33](#_Toc14509)

[PHỤ LỤC 34](#_Toc7518)

[TÀI LIỆU THAM KHẢO 35](#_Toc21518)

# DANH SÁCH CÁC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| STT | Từ viết tắt | Từ viết đầy đủ |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |

# DANH SÁCH HÌNH

[Hình 2.1. Quá trình roaming. 5](#_Toc19182)

[Hình 2.2. Kiến trúc SDN 12](#_Toc30985)

[Hình 3.1. Danh sách router được OpenWrt hỗ trợ 20](#_Toc27510)

[Hình 3.2. Phần mềm TFTPD64 20](#_Toc29729)

[Hình 3.3. Mô hình mạng trong nghiên cứu 21](#_Toc30655)

[Hình 3.4. Giao diện phần mềm PuTTY 22](#_Toc7097)

[Hình 3.5. Các giao diện nối với cầu nối của OpenVSwitch 23](#_Toc19514)

[Hình 3.6. Luồng tương tác của website 26](#_Toc10906)

[Hình 3.7. Chức năng của website 27](#_Toc12358)

[Hình 3.8. Hình trạng mạng thực tế 29](#_Toc11150)

[Hình 3.9. Màn hình xử lý cân bằng tải 30](#_Toc2239)

[Hình 3.10. Điều khiển luồng dữ liệu trên gateway 30](#_Toc26683)

[Hình 3.11. Màn hình quản lý chung trên website 31](#_Toc30734)

[Hình 3.12. Cảnh báo quá tải 31](#_Toc17268)

[Hình 3.13. Thông tin cơ bản trên router 32](#_Toc8084)

[Hình 3.14. Thông tin giao diện Wi-Fi của router 32](#_Toc876)

[Hình 3.15. Chức năng reboot 33](#_Toc5551)

# DANH SÁCH BẢNG

[Bảng 3-1. Các bước nghiên cứu 18](#_Toc170)

# TÓM TẮT

Mạng không dây Wi-Fi đang là loại mạng phổ biến đối với người dùng hiện nay. Đối với các doanh nghiệp hay cá nhân thì việc triển khai mạng không dây là vô cùng quan trọng bởi vì nó cung cấp tiện ích rất lớn đó là kết nối các thiết bị di động mà không cần sử dụng cáp nối nhờ vào các điểm truy cập không dây. Tuy nhiên, đối với các điểm truy cập đông người thì nó lại phát sinh những khó khăn vì lượng lớn người truy cập gây ra sự quá tải ở điểm truy cập.

Trên thực tế, hệ thống cân bằng tải cho Wi-Fi đã có nhưng phải đầu tư với chi phí khá cao nên ở luận văn này, tôi muốn hướng đến việc phát triển một hệ thống Wi-Fi cân bằng nhưng với một chi phí hợp lí.

Chúng tôi đưa ra những thuật toán, những thiết bị để hoàn thành tốt dự án được đưa ra, góp phần nghiên cứu để cải thiện vào hệ thống mạng Wi-Fi được tốt hơn.

Từ khóa: *Cân bằng tải. Wi-fi.*

# ABSTRACT

Wi-Fi is becoming more common among today's users. For businesses or individuals, deploying a wireless network is extremely important because it provides a large utility that connects mobile devices without the need for a cable connection using the access points. However, for crowded places, it is difficult because of the large number of clients causes overload at the access point.

In fact, the load balancing system for Wi-Fi is available, but it costs a lot to invest. In this thesis, I want to develop a balanced Wi-Fi system but at a reasonable cost.

We provide algorithms, devices to successfully complete the proposed project, contributing to improved research into the Wi-Fi network.  
Keywords: *load balancing. WIFI.*

# CHƯƠNG 1: TỔNG QUAN

## **ĐỘNG CƠ NGHIÊN CỨU**

Wi-Fi đã trở thành thiết bị quan trọng hàng đầu trong việc sử dụng mạng internet. Wi-Fi được cài đặt ở hầu hết các địa điểm công cộng và ở cà nhà riêng nó đang phát triển với một tốc độ nhanh chóng. Sử dụng Wi-Fi mang lại sự thuận tiện dễ dàng cho mọi người trong việc truy cập internet tuy nhiên việc sự dụng mạng Wi-Fi vào các điểm truy cập công cộng có quy mô lớn như khuôn viên trường đại học, quán cafe hay các công ty, xí nghiệp, siêu thị... đang gặp một số vấn đề cần giải quyết như:

* Khó theo dõi được trạng thái hoạt động của hệ thống Wi-Fi vì hầu hết các router do nhà cung cấp bị giới hạn thao tác đối với người dùng..
* Tải Wi-Fi không cân bằng: Thông thường đa số các thiết bị sử dụng Wi-Fi có xu hướng kết nối với router Wi-Fi đầu tiên, do các router có sóng khá mạnh nên sau khi kết nối, dù có đi khá xa các thiết bị vẫn không ngừng kết nối với router đầu tiên. Điều này dẫn đến một số Wi-Fi sẽ quá tải do số lượng nối kết quá nhiều còn một số khác thì dư tải và ít nối kết việc này ảnh hưởng đến trải nghiệm truy cập của mọi người và băng thông chung của hệ thống.

Để giải quyết các vấn đề trên đã có một số nghiên cứu cân bằng tải cho Wi-Fi cũng như có một số nhà sản xuất tạo ra các router cân bằng tải dành riêng cho các điểm truy cập Wi-Fi quy mô lớn tuy nhiên thông thường giá của các router này khá đắt đỏ.

Vì các lý do kể trên nhóm quyết định thực hiện đề tài “cân bằng tải Wi-Fi bằng hệ thống điều khiển tập trung” sử dụng các tiện ích trên firmware OpenWrt cũng như bộ điều khiển để tạo ra một mạng Wi-Fi có thể điều khiển bằng phần mềm, có thể “giao tiếp được với nhau” và có giá cả hợp lý.

### **MỤC TIÊU**

Xây dựng hệ thống Wi-Fi giá rẻ có khả năng điều khiển bằng bộ điều khiển tập trung. Đồng thời dựng trang web theo dõi hiệu năng, danh sách các thiết bị nối kết của router.

Đề xuất một giải thuật để giải quyết các vấn đề tải không cân bằng nêu trên để đảm bảo được thông lượng chung của hệ thống và trải nghiệm mạng một cách tốt nhất của người dùng.

### **NHỮNG NGHIÊN CỨU LIÊN QUAN**

Dựa theo bài báo khoa học tự đề “SAMF: An SDN-Based Framework for Access Point Management in Large-scale Wi-Fi Networks”[1] nghiên cứu về việc quản trị mạng Wi-Fi mạng cở lớn.

### **PHƯƠNG PHÁP NGHIÊN CỨU**

Đề tài là một phạm vi khá rộng và có những hướng đi phát triển tốt trong tương lai. Người thực hiện cần nắm kiến thức cơ bản về mạng không dây, xử lý ngôn ngữ lập trình cũng như những vấn đề xấu phát sinh mắc phải, không những thế người thực hiện cần hiểu rõ những khái niệm, những đặc điểm của cái thiết bị mạng không dây,… Để hoàn thành tốt việc nghiên cứu đề tài, phương pháp được chia ra như sau:

* Nghiên cứu về firmware OpenWrt, các API và các gói tiện ích trên OpenWrt.
* Nghiên cứu về kiến trúc Restful, cách sử dụng truy vấn.
* Xây dựng nhánh mạng
* Nghiên cứu công nghệ SDN

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## MẠNG WI-FI

### Mạng Wi-Fi là gì?

Wi-Fi viết tắt từ Wireless Fidelity hay mạng 802.11 là hệ thống mạng không dây sử dụng sóng vô tuyến, giống như điện thoại di động, truyền hình và radio. Hầu hết các thiết bị điện tử ngày nay như điện thoại di động, laptop… đều có thể kết nối Wi-Fi[2]

Hệ thống Wi-Fi hiện nay đang trở nên rất phổ biến ở các sân bay, quán cafe, thư viện khách sạn. Hệ thông cho phép người sử dụng truy cập internet ở các khu vực có sóng và hoàn toàn không cần dùng đến cáp nối. Ngoài các điểm kết nối hostposts Wi-Fi hiện nay có thể thiết lập tại nhà riêng tạo sự thuận tiện dễ dàng cho người sử dụng.

Hệ thống Wi-Fi truyền và phát tín hiệu ở tần số 2.4 GHz, 5 GHz và 60Ghz . Tần số này cao hơn so với các tần số sử dụng cho điện thoại di động, các thiết bị cầm tay và truyền hình. Tần số cao hơn cho phép tín hiệu mang theo nhiều dữ liệu hơn Hiện nay hệ thống mạng Wi-Fi đang có 5 chuẩn thông dụng là a,b,g,n,ac mỗi chuẩn hoạt động ở một băng tần nhất định và hỗ trợ tốc độ riêng.

**Chuẩn 802.11**:Năm 1997, IEEE đã giới thiệu chuẩn mạng không dây đầu tiên và đặt tên nó là 802.11. Tuy nhiên chuẩn 802.11 chỉ hỗ trợ tốc độ mạng tối đa 2 Mbps với băng tần 2.4 GHz, rất chậm so với ngày nay và không được áp dụng rộng rãi trên thị trường

**Chuẩn 802.11b**: Tháng 7 năm 1999, chuẩn 802.11b ra đời và hỗ trợ tốc độ lên đến 11Mpbs thay vì 2Mbps như trước kia. Tương tự thế hệ đầu tiên, chuẩn kết nối 802.11b cũng sử dụng băng tần 2.4 GHz rất dễ bị gây nhiễu từ các thiết bị điện tử khác như điện thoại di động, lò vi sóng,...

**Chuẩn 802.11a**: Được phát triển song song với chuẩn 802.11b, tuy nhiên chuẩn a thường được sử dụng trong các mạng của doanh nghiệp thay vì gia đình như chuẩn b vì giá thành khá cao. So với chuẩn 802.11b, chuẩn này hỗ trợ tốc độ tối đa gần gấp 5 lần, lên đến 54 Mpbs và sử dụng băng tần vô tuyến 5 GHz có thể tránh tình trạng bị nhiễu do các thiết bị khác. Tuy nhiên do tần số cao hơn nên phạm vi hoạt động của chuẩn 802.11a có phần hẹp hơn (40-100m) và khó xuyên qua các vật cản, vách tường.

**Chuẩn 802.11g**: Năm 2003, chuẩn Wi-Fi thế hệ thứ 3 ra đời được đặt là chuẩn 802.11g, chuẩn Wi-Fi này thậm chí còn được sử dụng ở nhiều mạng Wi-Fi các gia đình hiện nay. Chuẩn 802.11g được xem là kết hợp giữa chuẩn a và b trước kia, với giá thành khá rẻ (tuy có phần đắt hơn chuẩn b). Chuẩn 802.11g hỗ trợ tốc độ đến 54 Mpbs như chuẩn a nhưng sử dụng băng tần 2.4 GHz như chuẩn b, vì vậy chuẩn này có tốc độ cao, phạm vi tín hiệu tốt (80-200m).Và tất nhiên chuẩn này cũng có nhược điểm như chuẩn b là dễ bị nhiễu từ các thiết bị phát sóng khác. Do sự giống nhau về nhiều thông số, chuẩn kết nối 802.11g có khả năng tương thích ngược với chuẩn 802.11b và ngược lại.

**Chuẩn 802.11n**: Đây là chuẩn tương đối mới và đang sử dụng khá phổ biến hiện nay. Chuẩn Wi-Fi 802.11n được đưa ra nhằm cải thiện chuẩn 802.11g bằng cách sử dụng công nghệ MIMO (Multiple-Input Multiple-Output) tận dụng nhiều anten hơn. Chuẩn kết nối 802.11n hỗ trợ tốc độ tối đa lên đến 600 Mpbs, có thể hoạt động trên cả băng tần 2,4 GHz và 5 GHz, nếu router hỗ trợ thì hai băng tần này có thể cùng phát sóng song song.Chuẩn kết nối này đã và đang dần thay thế chuẩn 802.11g với tốc độ cao, phạm vi tín hiệu rất tốt (từ 100-250m) và giá thành đang ngày càng phù hợp với túi tiền người tiêu dùng.

**Chuẩn 802.11ac**: Chuẩn 802.11ac là chuẩn mới Wi-Fi nhất của IEEE đã được tung ra thị trường, áp dụng công nghệ đa anten đã có trên chuẩn 802.11n, với băng tần 5 GHz và tốc độ tối đa lên đến 1,3 Gigabit/giây người dùng sẽ trải nghiệm tốc độ mạng ở mức cao nhất.

**Chuẩn 802.11ad**: chuẩn 802.11ad phát ở tần số 60 GHz nhanh hơn so với chuẩn 802.11ac, tốc độ truyền dữ liệu tối đa đạt đến 4,6 Gigabit/giây

### Nguyên tắc hoạt động của mạng Wi-Fi

Bộ chuyển tín hiệu không dây (adapter) của thiết bị sử dụng mạng không dây sẽ chuyển đổi dữ liệu của nó sang tín hiệu vô tuyến và phát tín hiệu này đi bằng một ăng-ten.

Thiết bị Router Wi-Fi nhận những tín hiệu không dây của thiết bị sử dụng Wi-Fi gửi và giải mã chúng sau đó gửi qua mạng bằng các kết nối có dây (ethernet).

Hai quá trình trên sẽ hoạt động với chiều ngược lại để truyền tín hiệu từ mạng xuống thiết bị sử dụng Wi-Fi..

### Roaming trong mạng Wi-Fi

Roaming là quá trình xử lý, đảm bảo kết nối Wi-Fi của client khi di chuyển từ AP này sang AP khác. Khi ta di chuyển giữa các AP, kết nối phải được thiết lập trên AP mới. Vì vậy bất kỳ dữ liệu nào của ta đang gửi trước khi roaming sẽ được gửi lại từ AP cũ đến AP mới. Điều này giúp việc gửi nhận dữ liệu trong và sau roaming không bị mất. Khoảng thời gian trì hoãn là không đáng kể.

Để có thể tiến hành roaming, các AP cần phải cùng chuẩn, SSID, chế độ mã hóa/ xác thực nhưng không được trùng kênh. Đầu tiên, router Wi-Fi thứ nhất phải được cấu hình không trùng kênh với router Wi-Fi thứ hai. Điều này giúp máy tính của ta phân biệt được tín hiệu của router Wi-Fi thứ nhất và router Wi-Fi thứ hai. Chẳng hạn, 2 router Wi-Fi cùng hỗ trợ chuẩn 802.11g, ta có thể cấu hình router Wi-Fi thứ nhất sử dụng kênh số 1 và router Wi-Fi thứ hai sử dụng kênh 6.

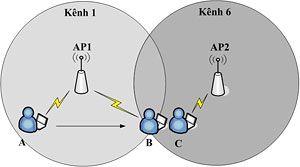
Việc xử lý roaming sẽ do trình điều khiển card Wi-Fi của client đảm nhận (AP không tham gia vào quá trình này). Client có thể thực hiện một trong hai cách để chuẩn bị tiến hành roaming:

* Cách 1: khi client nhận thấy tín hiệu từ router giảm đi, nó sẽ thực hiện tìm kiếm router khác gần nhất.
* Cách 2: khi client di chuyển ra cận vùng phủ sóng router, tín hiệu yếu, không đảm bảo kết nối, nó sẽ tiến hành tìm kiếm router khác gần nhất.

Trình điều khiển card Wi-Fi của client sẽ quyết định lúc nào cần thực hiện roaming. Giải thuật roaming là các giải thuật bí mật của nhà sản xuất card Wi-Fi, ta không thể nhìn thấy thời điểm chính xác khi nào diễn ra tiến trình roaming. Giải thuật roaming thường căn cứ vào cường độ tín hiệu, chất lượng tín hiệu...

Với mỗi thiết bị di động khác nhau sẽ có thời điểm roaming khác nhau. Một số thì tiến hành roaming ngay khi vào vùng tín hiệu mới; một số thì lại roaming khi nhận thấy tín hiệu AP mới tốt hơn. Khi client quyết định tiến hành roaming thì bước tiếp theo là nó phải tìm kiếm, dò kênh AP mới. Có 02 cách:

* Dò tìm thụ động: client vừa dò kênh, vừa lắng nghe báo hiệu 802.11 từ router mới.
* Dò tìm chủ động: client vừa dò kênh, vừa gửi frame thăm dò 802.11 để tìm kiếm router mới.

Khi client dò tìm thụ động, nó chỉ chờ nhận báo hiệu từ AP. Dò tìm chủ động, client phải gửi thăm dò và chờ nhận tín hiệu trả lời của AP. Dò tìm chủ động sẽ hiệu quả hơn vì client trực tiếp tìm kiếm AP, giúp rút ngắn thời gian trì hoãn việc gửi nhận dữ liệu.

1. Quá trình roaming.

Vì client không thể kết nối với 2 router cùng lúc nên sau khi tiến hành roaming, dò tìm kênh, client sẽ ngắt kết nối với router thứ nhất và chuyển qua với router thứ hai. Kết nối thành công, client sẽ sử dụng router thứ hai làm điểm truy cập mới. [2]

## FIRMWARE OPENWRT

### Firmware OpenWrt là gì?

OpenWrt là một hệ điều hành nhúng dựa trên nhân Linux, và thường được nhúng trên các thiết bị nhúng vào mạng lưới định tuyến đường truyền. Thành phần chính của OpenWrt là nhân Linux, util-linux, uClibc và BusyBox. Các thành phần của OpenWrt đã được tối ưu kích thước để có thể nhúng vào bộ nhớ có hạn của thiết bị router dùng trong gia đình. Vì được xây dựng dựa trên nhân Linux nên OpenWrt có thể biến một router của nhà cung cấp với ít khả năng thao tác của người dùng trở nên linh hoạt hơn trong việc quản lý và điều khiển

OpenWrt được cấu hình bằng cách sử dụng giao diện dòng lệnh (ash Shell), hoặc một giao diện Web ([Luci](http://wiki.openwrt.org/doc/techref/luci)). Có khoảng 3.500 gói phần mềm tùy chọn có sẵn để cài đặt qua hệ thống quản lý gói [opkg.](https://vi.wikipedia.org/w/index.php?title=Opkg.&action=edit&redlink=1" \o "Opkg. (trang chưa được viết))

OpenWrt có thể chạy trên các bộ định tuyến CPE, hệ thống mạng nhỏ, điện thoại thông minh (ví dụ: Neo FreeRunner), máy tính bỏ túi (ví dụ: Ben NanoNote), thậm chí cả máy tính xách tay (ví dụ: One Laptop per Child (OLPC)). Ngoài ra, OpenWrt có thể chạy trên các máy tính thông thường (ví dụ như kiến trúc [x86).](https://vi.wikipedia.org/wiki/X86" \o "X86) Nhiều bản vá lỗi từ các mã nguồn cơ sở của OpenWrt đã được đưa và đóng góp ngược lại cho nhân Linux bằng đường chính.

### Những tính năng của OpenWrt

OpenWrt có sự mở rộng rất linh hoạt và đa dạng tính năng. Các tính năng chính:

* OpenWrt cũng cấp một hệ thống tập tin gốc cho phép người dùng có thể thêm, xóa hoặc sửa đổi bất kỳ tập tin. Điều này có thể thực hiện bằng việc sử dụng overlayfs có thệ thống tập tin chỉ đọc được nén với SquashFS và JFFS2 có thể ghi Copy-on-Write.
* Quản lý gói opkg, cho phép người dùng cài đặt và gỡ bỏ phần mềm. Trong các kho có chứa khoảng 3500 gói.
* Một tập hợp các mã gọi là UCI (giao diện cấu hình Unified) Dự kiến để thống nhất và đơn giản hóa cấu hình của toàn bộ hệ thống
* Cấu hình mở rộng của toàn bộ trình điều khiển phần cứng, ví dụ như xây dựng tích hợp switch mạng và [VLAN,](https://vi.wikipedia.org/wiki/Virtual_LAN" \o "Virtual LAN) WNIC s, modem DSL s, FX, các nút phần cứng có sẵn, vv
* Khả năng toàn diện để cấu hình các tính năng mạng như:
* Định tuyến thông qua iproute2, Quagga, BIRD, vv
* Chức năng [không dây,](https://vi.wikipedia.org/w/index.php?title=Kh%C3%B4ng_d%C3%A2y,&action=edit&redlink=1" \o "Không dây, (trang chưa được viết)) ví dụ như làm những hành động thiết bị như một repeater không dây, một điểm truy cập không dây, một cầu nối không dây, một cổng khóa,…
* Bảo mật không dây: Packet injection, ví dụ: [Airpwn](http://airpwn.sourceforge.net/Airpwn.html), lorcon,…
* Tường lửa, NAT và port forwarding thông qua netfilter; PeerGuardian có sẵn
* Tự động được cấu hình cổng giao thức chuyển tiếp.
* IP tunneling
* Domain Name System (DNS) và [DHCP](https://vi.wikipedia.org/wiki/DHCP" \o "DHCP) qua dnsmasq, MaraDNS, vv
* Dịch vụ DNS động để duy trì một tên miền cố định với một ISP không cung cấp một địa chỉ IP tĩnh
* hệ thống phân phối không dây (WDS) bao gồm WPA-PSK, WPA2-PSK, WPA-PSK / WPA2-PSK Mixed-Mode chế độ mã hóa
* OpenWrt hỗ trợ bất kỳ phần cứng có hỗ trợ Linux như máy in, webcam, card âm thanh
* Mở rộng một giao diện Web [Ajax,](https://vi.wikipedia.org/w/index.php?title=Ajax,&action=edit&redlink=1" \o "Ajax, (trang chưa được viết)) nhờ dự án Luci
* Sửa lỗi và cập nhật thường xuyên, ngay cả đối với các thiết bị không còn được hỗ trợ bởi các nhà sản xuất của họ.

**Dự án Luci**

Luci được tạo ra vào tháng 3 năm 2018 với tên gọi là FFLuci với mục tiêu là tạo ra giao diện web thân thiện, có thể mở rộng và bảo trì cho các thiết bị nhúng sử dụng firmware OpenWrt. Trong khi các giao diện cấu hình khác sử dụng những ngôn ngữ kịch bản nặng nề thì Luci sử dụng ngôn ngữ lập trình là lua và chia giao diện thành các phần logic (model và view) sử dụng các thư viện hướng đối tượng để đảm bảo hiệu suất cao hơn, kích thước cài đặt nhỏ hơn, thời gian chạy nhanh hơn và khả năng bảo trì tốt hơn.

**JSON RPC API**

JSON-RPC hoạt động bằng cách gửi yêu cầu tới máy chủ triển khai giao thức này. Các client trong trường hợp này thường là phần mềm có ý định gọi một phương pháp duy nhất của một hệ thống từ xa. Nhiều tham số đầu vào có thể được chuyển tới phương thức từ xa như một mảng hoặc đối tượng, trong khi chính phương thức đó cũng có thể trả về nhiều dữ liệu đầu ra. (Điều này tùy thuộc vào phiên bản được triển khai.)

Tất cả các kiểu truyền là các đối tượng đơn lẻ, được tuần tự hóa bằng JSON. Một yêu cầu là một cuộc gọi đến một phương pháp cụ thể được cung cấp bởi một hệ thống từ xa. Nó phải chứa ba thuộc tính nhất định:

* method - Một chuỗi với tên của phương thức được gọi.
* params - Một đối tượng hoặc mảng của các giá trị được chuyển thành các tham số cho phương thức đã định nghĩa.
* id - Giá trị của bất kỳ loại nào được sử dụng để khớp với phản hồi với yêu cầu mà nó đang trả lời.

Người nhận yêu cầu phải trả lời bằng phản hồi hợp lệ cho tất cả các yêu cầu đã nhận. Phản hồi phải chứa các thuộc tính :

* result - Dữ liệu được trả về bởi phương thức được dẫn. Nếu xảy ra lỗi khi gọi phương thức, giá trị này phải là rỗng.
* error - Một mã lỗi được chỉ định nếu có lỗi khi gọi phương thức, nếu không thì sẽ là null.
* id - Id của yêu cầu mà nó đang phản hồi.

OpenWrt cung cấp JSON-RPC-API cho phép lập trình viên có thể thông qua kiến trúc Restful đưa ra các lời gọi hàm và nhận về các dữ liệu, thông tin mong muốn.

Dựa trên tài liệu cung cấp ở site: [https://github.com/OpenWrt/luci/wiki/JsonRpcHowTo[3],](https://github.com/OpenWrt/luci/wiki/JsonRpcHowTo,) Lập trình viên có thể xây dựng các function để theo dõi thông số của các router, theo dõi phiên kết nối của các client cũng như việc cập nhật cấu hình bằng những lời gọi hàm thích hợp.

Tùy mục đích các API cung cấp bởi OpenWrt được chia theo kiến trúc như sau:

Mỗi một thư viện của JSON-RPC-API sẽ được đưa vào một đường dẫn Restful riêng với những chức năng được cung cấp sẵn.

VD: Để thực hiện lệnh khởi động lại router bằng thư viện luci.sys ta có thể thực hiện với kiến trúc restful sau:

curl http://<hostname>/cgi-bin/luci/rpc/sys?auth=CHUOITOKEN '

{

"id": 1,

"jsonrpc": "2.0",

"method": "reboot",

"params": []

}'

chú thích:CHUOITOKEN

Chuỗi token là chuỗi xác thực được cung cấp trong một phiên làm việc để lập trình viên có quyền kiểm soát đối với toàn bộ các thư viện Json-Rpc. Nếu không có chuỗi này các lời gọi hàm Restful sẽ không thực thi được.

Để lấy chuỗi token ta có thể sử dụng kiến trúc Restful sau:

curl http://<hostname>/cgi-bin/luci/rpc/auth --data '

{

"id": 1,

"method": "login",

"params": [

"youruser",

"somepassword"

]

}'

**UCI**

UCI là viết tắt của Unified Configuration Interface và là một hệ thống tập trung cấu hình các dịch vụ OpenWrt.[4]

UCI kế thừa việc cấu hình dựa trên NVRAM được tìm thấy trong loạt OpenWrt trắng của Nga và là giao diện người dùng cấu hình chính cho các thiết lập hệ thống quan trọng nhất. Ví dụ như cấu hình giao diện mạng chính, cài đặt không dây, chức năng ghi nhật ký và cấu hình truy cập từ xa. Ngoài ra, nhiều gói trong kho lưu trữ OpenWrt đã được thực hiện tương thích với hệ thống UCI.

Các ứng dụng được tạo tương thích UCI bằng cách chỉ cần viết tập tin cấu hình ban đầu (được chương trình đọc) theo cài đặt đã chọn trong tập tin UCI tương ứng. Điều này được thực hiện khi chạy các kịch bản khởi tạo trong /etc/init.d/. Do đó, khi bắt đầu một daemon với tập lệnh khởi tạo tương thích UCI, nên biết rằng tập tin cấu hình ban đầu của chương trình bị ghi đè. Ngoài ra, tập tin cấu hình của ứng dụng thường được lưu trữ trong RAM thay vì trong flash, bởi vì nó không cần phải được lưu trữ trong bộ nhớ không mất và nó được viết lại sau mỗi thay đổi, dựa trên tập tin UCI. Có nhiều cách để vô hiệu hóa UCI trong trường hợp ta muốn điều chỉnh các thiết lập trong tập tin cấu hình ban đầu không có sẵn thông qua UCI, trong cifs.server, có thể xem cách tắt UCI cho samba chẳng hạn.

Đối với những chương trình không tương thích UCI, có một danh sách tiện lợi của một số tập tin cấu hình không phải UCI mà ta có thể muốn. Lưu ý rằng, đối với hầu hết các chương trình của bên thứ ba, ta nên tham khảo tài liệu riêng của chương trình.  
Nguyên tắc chung.

Cấu hình trung tâm của OpenWrt được chia thành nhiều tập tin nằm trong thư mục /etc/config/. Mỗi tập tin liên quan đến một phần của hệ thống mà nó cấu hình. Ta có thể chỉnh sửa các tập tin cấu hình bằng trình soạn thảo văn bản hoặc sửa đổi chúng bằng chương trình tiện ích dòng lệnh uci. Các tập tin cấu hình UCI cũng có thể sửa đổi thông qua các API lập trình khác nhau (như Shell, Lua và C), cũng là cách các giao diện web như LuCI thay đổi các tập tin UCI.

Khi thay đổi tập cấu hình UCI, cho dù thông qua trình soạn thảo văn bản hoặc dòng lệnh, dịch vụ hoặc tập thực thi bị ảnh hưởng phải được khởi động lại (hoặc, trong một số trường hợp) bằng lệnh init.d, sao cho Cấu hình UCI cập nhật được áp dụng cho chúng. Nhiều chương trình được thực hiện tương thích với UCI theo cách này bằng cách làm cho init.d script của họ viết các tập cấu hình phần mềm tiêu chuẩn cụ thể của họ. Tập lệnh init.d đầu tiên viết đúng một tập tin cấu hình vào vị trí dự kiến của phần mềm và nó được đọc lại bằng cách khởi động lại tập tin thực thi. Lưu ý rằng chỉ bắt đầu thực thi trực tiếp, không có các cuộc gọi init.d, sẽ không dẫn đến một bản cập nhật UCI để relegate cấu hình UCI thành tập tin cấu hình dự kiến của chương trình. Những thay đổi trong các tập tin trong /etc/config/ sau đó không có hiệu lực.

**Hostpad**

Hostapd là một daemon không gian người dùng cho các máy chủ xác thực và điểm truy cập. Nó thực hiện quản lý điểm truy cập IEEE 802.11, Trình xác thực IEEE 802.1X / WPA / WPA2 / EAP, máy khách RADIUS, máy chủ EAP và máy chủ xác thực RADIUS. Phiên bản hiện tại hỗ trợ Linux (Trình điều khiển máy chủ AP, madwifi, trình điều khiển dựa trên mac80211) và FreeBSD (net80211).

Hostapd được thiết kế để trở thành một chương trình "daemon" chạy trong nền và hoạt động như thành phần phụ trợ kiểm soát xác thực. hostapd hỗ trợ các chương trình giao diện người dùng riêng biệt và một giao diện văn bản ví dụ, hostapd\_cli, được bao gồm với hostapd[5]

**Ubus**

Để cung cấp thông tin liên lạc giữa các trình tiện ích và ứng dụng khác nhau trong OpenWrt, một dự án được gọi là ubus đã được phát triển. Nó bao gồm một số phần bao gồm daemon, thư viện và một số người trợ giúp thêm.

Trung tâm của dự án này là daemon ubusd. Nó cung cấp một giao diện cho các trình tiện ích khác để tự đăng ký cũng như gửi tin nhắn. Đối với những người tò mò, giao diện này được thực hiện bằng cách sử dụng socket Unix và nó sử dụng thông điệp TLV(type-length-value) .

Để đơn giản hóa việc phát triển phần mềm bằng cách sử dụng ubus (kết nối với nó) một thư viện có tên libubus đã được tạo ra.

Mỗi daemon đăng ký một tập hợp các đường dẫn dưới một không gian tên cụ thể. Mỗi đường dẫn có thể cung cấp nhiều thủ tục với bất kỳ số lượng đối số nào. Thủ tục có thể trả lời bằng tin nhắn.

## TFTP

TFTP là viết tắt của Trivial File Transfer Protocol là một giao thức truyền tập tin nhỏ gọn cho phép client lấy tập tin hoặc đặt tập tin lên máy chủ từ xa. Một trong những ứng dụng chính của nó là trong giai đoạn đầu của các nút khởi động từ một mạng cục bộ. TFTP đã được sử dụng cho ứng dụng này bởi vì nó rất đơn giản để thực hiện.

TFTP được chuẩn hóa lần đầu tiên vào năm 1981 và đặc điểm kỹ thuật hiện tại cho giao thức có thể được tìm thấy trong RFC 1350.

## TIẾN TRÌNH KHỞI ĐỘNG ROUTER

Khi khởi động router, nó sẽ trải qua tiến trình khở động gồm 4 bước:

Bước 1 - POST (Power on self test): Router thực kiểm tra các module phần cứng (Hardware diagnostic)

Bước 2: Sau khi kiểm tra xong và đảm bảo mọi module hoạt động hợp lệ, router sẽ chạy chương trình bootstrap nằm trong bộ nhớ ROM để load chương trình bootstrap từ ROM vào bộ nhớ RAM

Bước 3: Sau khi đã load chương trình bootstrap vào RAM, chương trình sẽ load hệ điều hành IOS cho router để chạy theo tiến trình sau:

Đầu tiên, nếu trình flash có tồn tại IOS, router sẽ giải nén và load IOS đầu tiên để vận hành router.

Trong trường hợp không tìm thấy IOS hoặc IOS trong flash lỗi, router sẽ tiến hành tìm kiếm hệ điều hành từ các TFTP server xung quanh nó để load về và vận hành

Nếu cả hai tình huống trên đều không tìm được IOS thì router sẽ load hệ điều hành phụ trong ROM để chạy

Bước 4: router load tập tin cấu hình cho router để chạy cho nó.[6]

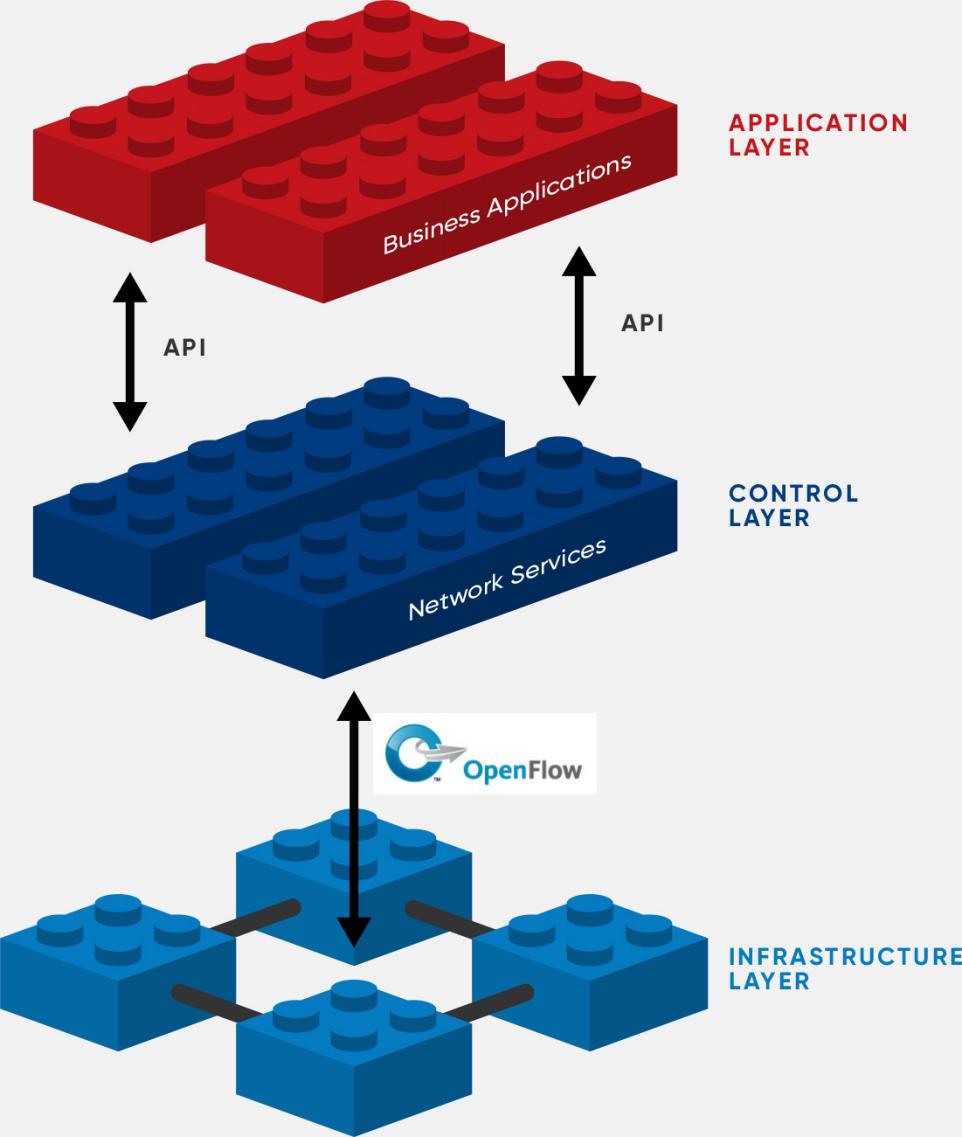
## GIỚI THIỆU CÔNG NGHỆ SDN

### SDN là gì?

Mạng điều khiển bằng phần mềm (SDN) là một kiến trúc mới nổi có tính năng động, dễ quản lý, hiệu quả về chi phí và có thể thích nghi, làm cho nó trở nên lý tưởng cho tính chất động, băng thông cao của các ứng dụng ngày nay. Kiến trúc này tách các chức năng điều khiển và chuyển tiếp mạng cho phép điều khiển mạng trở thành lập trình trực tiếp và cơ sở hạ tầng cơ bản được trừu tượng hóa cho các ứng dụng và dịch vụ mạng. Giao thức OpenFlow là một yếu tố cơ bản để xây dựng các giải pháp SDN.

Kiến trúc SDN gồm 3 lớp riêng biệt:

* Lớp ứng dụng (Application layer): là các ứng dụng được triển khai dựa trên việc kết nối với lớp điều khiển thông qua các API.
* Lớp điều khiển (Control layer): là nơi tập trung các bộ điều khiển thực hiện việc điều khiển và cấu hình mạng theo các yêu cầu từ lớp ứng dụng.
* Lớp cơ sở hạ tầng (Infrastructure layer): Là các thiết bị mạng thực tế (vật lý hay ảo hóa) thực hiện việc chuyển tiếp gói tin theo sự điều khiển của lớp điểu khiển. Một thiết bị mạng có thể hoạt động theo sự điều khiển của nhiều bộ điều khiển khác nhau, điều này giúp tăng cường khả năng ảo hóa của mạng.



1. Kiến trúc SDN

### Giao thức OpenFlow

OpenFlow là tiêu chuẩn cung cấp khả năng giao tiếp giữa các giao diện của lớp điều khiển và lớp cơ sở hạ tầng trong kiến trúc SDN. OpenFlow cho phép truy cập trực tiếp và điều khiển lớp cơ sở hạ tầng như switch, router cả thiết bị thật lẫn ảo. Nhờ có OpenFlow mà tách được lớp điều khiển với các thiết bị chuyển mạch thực tế.

Giao thức OpenFlow phải được triển khai trên cả hai thiết bị ở lớp điều khiển và lớp cơ sở hạ tầng. Việc sử dụng OpenFlow có thể thông qua các tập lệnh để điều khiển mặt phẳng cơ sở hạ tầng.

### OpenVSwitch

OpenVSwitch là một dự án chuyển mạch ảo hỗ trợ OpenFlow nhằm cung cấp lớp chuyển mạch cho môi trường ảo hóa phần cứng, nó cũng hỗ trợ nhiều tính năng cho phép đơn giản hóa việc cấu hình cho bộ chuyển mạch như:

* VLAN tagging và chuẩn 802.1q trunking
* Spanning tree protocol 802.1D
* Port mirroring (SPAN/RSPAN)
* Kiểm soát QoS

Các thành phần chính của OpenvSwitch:

* ovs-vswitchd: thực hiện chuyển đổi các luồng chuyển mạch. Công cụ tương tác: ovs-dpctl, ovs-appctl, ovs-ofctl, sFlowTrend
* ovsdb-server: là một lightweight database server, cho phép ovs-vswitchd thực hiện các truy vấn đến cấu hình. Công cụ tương tác: ovs-vsctl, ovsdb-client
* ovs-dpctl: công cụ để cấu hình các switch kernel module.
* ovs-vsctl: tiện ích để truy vấn và cập nhật cấu hình ovs-vswitchd.

## PYTHON

Python là ngôn ngữ lập trình bậc cao nhằm giúp cho việc lập trình đa năng. Cha đẻ của ngôn ngữ lập trình python là Guido van Rossum đã giới thiệu python vào năm 1991. Python được đánh giá là một ngôn ngữ mạnh, dễ đọc và có cú pháp đơn giản giúp người lập trình dễ sử dụng.

Ngôn ngữ python đã ra mắt phiên bản 3.x điều này chứng tỏ python là một ngôn ngữ đang có vị thế lớn và quan trọng trong giới lập trình.

## RYU FRAMEWORK

Ryu là một khung làm việc điều khiển mạng dựa trên các thành phần. Ryu cung cấp các thành phần phần mềm với API được xác định rõ, giúp các nhà phát triển dễ dàng tạo ra các ứng dụng quản lý và kiểm soát mạng.

Ryu hỗ trợ các giao thức khác nhau để quản lý các thiết bị mạng, như OpenFlow, Netconf, OF-config, vv. Về OpenFlow, Ryu hỗ trợ đầy đủ các phần mở rộng 1.0, 1.2, 1.3, 1.4, 1.5 và Nicira. Tất cả các mã đều có sẵn miễn phí theo giấy phép Apache 2.0. [7]

## CÔNG NGHỆ WEB

- World Wide Web, gọi tắt là Web hoặc WWW, mạng lưới toàn cầu là một không gian thông tin toàn cầu mà mọi người có thể truy cập (đọc và viết) qua các thiết bị kết nối với mạng Internet; một hệ thống thông tin trên Internet cho phép các tài liệu được kết nối với các tài liệu khác bằng các liên kết siêu văn bản, cho phép người dùng tìm kiếm thông tin bằng cách di chuyển từ tài liệu này sang tài liệu khác.

- Website còn gọi là trang web, là một tập hợp các trang web con, bao gồm văn bản, hình ảnh, video, flash v.v.. website chỉ nằm trong một tên miền hoặc tên miền phụ lưu trữ trên các máy chủ.

- Một website cần có những thành phần chính như tên miền là tên riêng trỏ đến máy chủ chứa các tập tin nguồn, hosting là máy chủ chứa các tập tin nguồn, mã nguồn là các tập tin html,xhml… Website được tương tác và hiển thị với người dùng qua trình duyệt web.

Các nền tảng để xây dựng website:

**HTML**: là chữ viết tắt của cụm từ HyperText Markup Language (dịch là Ngôn ngữ đánh dấu siêu văn bản) được sử dụng để tạo một trang web, trên một website có thể sẽ chứa nhiều trang và mỗi trang được quy ra là một tài liệu HTML (thi thoảng mình sẽ ghi là một tập tin HTML). Cha đẻ của HTML là Tim Berners-Lee, cũng là người khai sinh ra World Wide Web và chủ tịch của World Wide Web Consortium (W3C – tổ chức thiết lập ra các chuẩn trên môi trường Internet).

**CSS**: CSS là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (ví dụ như HTML). Nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,…thì CSS sẽ giúp chúng ta có thể thay đổi “diện mạo” của các phần tử HTML đó như đổi màu sắc trang, đổi màu chữ, thay đổi cấu trúc,…rất nhiều.

**PHP**: PHP được xây dựng ban đầu như là một dự án mã nguồn mở nhỏ, nhưng càng về sau người ta càng cảm thấy được sự hữu ích của nó nên đã phát triển php thành ngôn ngữ phổ biến như bây giờ. Tên ban đầu của ngôn ngữ php là php/fi, php/fi do Rasmus Lerdorf tạo ra năm 1994, ban đầu php/fi được xem như là một tập con đơn giản của các mã kịch bản Perl để theo dõi tình hình truy cập đến bản sơ yếu lý lịch của Rasmus Lerdorf trên mạng. Sau một thời gian Rasmus đã quyết định công bố mã nguồn của PHP/FI cho mọi người xem, sử dụng cũng như sửa các lỗi có trong nó, đồng thời cải tiến mã nguồn.

PHP là viết tắt của "PHP: Hypertext Preprocessor".

PHP là ngôn ngữ lập trình kịch bản viết cho máy chủ mà được nhúng trong HTML. Nó được sử dụng để quản lý nội dụng động, cơ sở dữ liệu, kiểm tra phiên làm việc...

PHP được tích hợp với một số Database thông dụng như MySQL, PostgreSQL, Oracle, Sybase, Informix, và Microsoft SQL Server.

PHP hỗ trợ một số lượng rộng rãi các giao thức lớn như POP3, IMAP, và LDAP. PHP4 bổ sung sự hỗ trợ cho Java và các cấu trúc đối tượng phân phối (COM và CORBA).

PHP có khả năng cực lớn trong việc lập trình web:

* PHP thực hiện được các hàm hệ thống, ví dụ: từ các file trên một hệ thống, nó có thể tạo, mở, đọc, ghi và đóng chúng.
* PHP có thể xử lý các form, ví dụ: thu thập dữ liệu từ file, lưu dữ liệu vào một file, thông qua email ta có thể gửi dữ liệu, trả về dữ liệu tới người dùng.
* Bạn có thể thêm, xóa, sửa đổi các phần tử bên trong Database của bạn thông qua PHP.
* Truy cập các biến Cookie và thiết lập Cookie.
* Sử dụng PHP có thể mã hóa dữ liệu theo các chuẩn khác nhau.

Một trong những tính năng ưu việt của php là hỗ trợ gửi và nhận các Restful request-công nghệ cốt lõi trong đề tài luận văn này.

**MySQL**: là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. MySQL miễn phí hoàn toàn cho nên bạn có thể tải về MySQL từ trang chủ. Nó có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS,...

MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).

MySQL được sử dụng cho việc bổ trợ PHP, Perl, và nhiều ngôn ngữ khác, nó làm nơi lưu trữ những thông tin trên các trang web viết bằng PHP hay Perl,…

**JavaScript**: Là một trong những ngôn ngữ lập trình chính cho lập trình web. Javascritp giúp điều chỉnh lại cách hoạt động của một trang web. Javascritp có khả năng liên kết trực tiếp với các thẻ html thay đổi nội dung cũng như cách hiển thị của chúng. Javascript có thể lắng nghe các sự kiện trên trang web thậm chí đén từng cú click chuột của người dùng với mọi thành phần của trang web. Nói tóm lại javascript mang một tiềm năng rất lớn giúp website tương tác tốt hơn đối với các khách hàng sử dụng. Mang nhiều ưu điểm tuy nhiên javascipt là một ngôn ngữ lập trình hoạt động ở phía client tức là script được tải về máy của khách truy cập và được xử lý tại đó thay vì phía server là xử lý trên server rồi mới đưa kết quả tới khách truy cập. Một số trình duyệt sẽ cho phép người dùng tắt bỏ đi javascript nên hãy lưu ý cách mà website sẽ hoạt động nếu nó không có javascritp.

**jQuery**: Là một thư viện kiểu mới của JavaScript, được tạo bởi John Resig vào năm 2006 với một phương châm tuyệt vời: Write less, do more - Viết ít hơn, làm nhiều hơn. jQuery làm đơn giản hóa việc truyền tải HTML, xử lý sự kiện, tạo hiệu ứng động và tương tác Ajax. Với jQuery, khái niệm RAPId Web Development đã không còn quá xa lạ. jQuery là một bộ công cụ tiện ích JavaScript làm đơn giản hóa các tác vụ đa dạng với việc viết ít code hơn.

**Bootstrap**: Là một framework cho phép thiết kế website reponsive nhanh hơn và dễ dàng hơn

Bootstrap là bao gồm các HTML templates, CSS templates và Javascript tạo ra những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác. Bootstrap có thêm các plugin Javascript trong nó. Giúp cho việc thiết kế reponsive dễ dàng hơn và nhanh chóng hơn.

**Công nghệ Restful**: Rest Là một kiều kiến trúc lập trình, nó định nghĩa các quy tắc để thiết kết các web service chú trọng vào tài nguyên hệ thống. Trong kiến trúc REST mọi thứ đều được coi là tài nguyên, chúng có thể là: tệp văn bản, ảnh, trang html, video, hoặc dữ liệu động… REST server cung cấp quyền truy cập vào các tài nguyên, REST client truy cập và thay đổi các tài nguyên đó. Ở đây các tài nguyên được định danh dựa vào URI, REST sử dụng một vài đại diện để biểu diễn các tài nguyên như văn bản, JSON, XML.

Bốn quy tắc cơ bản để cài một Restful webservice:

* Sử dụng một cách rõ ràng các phương thức http
* Để tạo một tài nguyên trên server ta dùng phương thức POST.
* Để lấy(đọc) tài nguyên trên server ta dùng phương thức GET.
* Để update tài nguyên trên server ta dùng phương thức PUT.
* Để xóa tài nguyển trên server ta dùng phương thức DELETE.
* Phi trạng thái: Theo REST, trạng thái hoặc được giữ trên client hoặc được chuyển thành trạng thái của tài nguyên. Nói một cách khác một server sẽ không bao giờ giữ trạng thái thông tin trao đổi với bất kỳ client nào nó giao tiếp, mỗi request lên server thì client phải đóng gói thông tin đầy đủ để thằng server hiểu được. Điều này giúp hệ thống của bạn dễ phát triển,bảo trì, mở rộng vì không cần tốn công CRUD trạng thái của client. Ngoài ra còn một nguyên nhân quan trọng hơn đó là nó tách biệt client khỏi sự thay đổi của server.

Cấu trúc thư mục như URI: URI trong RESTful web service phải tự mô tả, hoặc tham chiếu được cái mà nó trỏ tới và các tài nguyên liên quan. Ngoài ra URI cũng phải đơn giản, có thể đoán biết được, và dễ hiểu. Để tạo ra URI với yêu cầu trên thì ta nên định nghĩa URI có câu trúc giống thư mục. Loại URI này có phân cấp, có gốc là một đường dẫn đơn, các nhánh từ gốc là các đường dẫn phụ dẫn đến các các vùng service chính.

* Chuyển đổi XML, JSON hoặc cả hai: Điều cuối cùng trong tập các ràng buộc khi thiết kế RESTful web service phải làm là định dạng dữ liệu mà ứng dụng và service trao đổi trong phụ tải request/response hoặc trong HTTP body. Cung cấp nhiều đại diện biểu diễn cho tài nguyên cho các request khác nhau. Cụ thể ở đây ta có thể sử dụng các một vài kiểu MIME thông dụng sau:
* JSON
* XML
* Điều này cho phép các service sử dụng bởi các client viết bởi các ngôn ngữ khác nhau, chạy trên nhiều nền tảng và thiết bị khác nhau. Sử dụng các kiểu MIME cho phép client chọn dạng dữ liệu phù hợp với nó.

# CHƯƠNG 3: NỘI DUNG VÀ KẾT QUẢ NGHIÊN CỨU

## NỘI DUNG NGHIÊN CỨU

Trong đề tài này, chúng tôi dựa theo quy trình trong bảng … để nghiên cứu và hoàn thiện sản phẩm

1. Các bước nghiên cứu

|  |  |
| --- | --- |
| Bước 1: Tìm hiều thông tin | Tìm hiểu các thông tin về phần cứng, firmware OpenWrt |
| Bước 2:  Xây dựng hệ thống | Xây dựng bộ điều khiển, cài đặt giải thuật cân bằng tải, xem xét và cải tiến giải thuật  **Xây dựng website**  Lựa chọn, xây dựng mô hình mạng ban đầu  Cài đặt firmware cho router  **Chọn controller**  Xây dựng các hàm chức năng, hàm điều khiển cho mô hình mạng đã xây dựng |
| Bước 3: Đánh giá và kết luận | Đánh giá giải thuật, hiệu năng hệ thống  Kết luận |

Cụ thể theo quy trình trên chúng tôi bắt đầu nghiên cứu bằng việc tìm hiểu những thiết bị router nào là phù hợp với đề tài bởi vì theo hướng dẫn từ website của OpenWrt (https://openwrt.org), không phải loại router nào trên thị trường đều có thể sử dụng firmware OpenWrt. Chính vì lí do đó, chúng tôi tiến hành nghiên cứu những thiết bị router nào trên thị trường có thông tin phần cứng đáp ứng được nhu cầu nhưng phải có giá cả hợp lý để triển khai cho luận văn này.

Tiếp theo đó, chúng tôi tiến hành cài đặt firware cho các thiết bị router và lựa chọn một hình trạng của một nhánh mạng thường được triển khai để tiến hành nghiên cứu trong đề tài này. Sau khi đã chọn được hình trạng mạng, chúng tôi tiến hành nghiên cứu các lý thuyết về mạng Wi-Fi và các nghiên cứu liên quan đến đề tài để đề xuất ra một giải thuật nhằm giải quyết vấn đề tải không cân bằng của hệ thống. Theo đó, chúng tôi lựa chọn một bộ điều khiển và xây dựng các chức năng để giải quyết vấn đề đã đặt ra. Sau khi đã xây dựng được bộ điều khiển, chúng tôi mới xây dựng một website để theo dõi các thông tin quan trọng của hệ thống router.

Cuối cùng, chúng tôi đánh giá lại giải thuật và hoàn thiện sản phẩm.

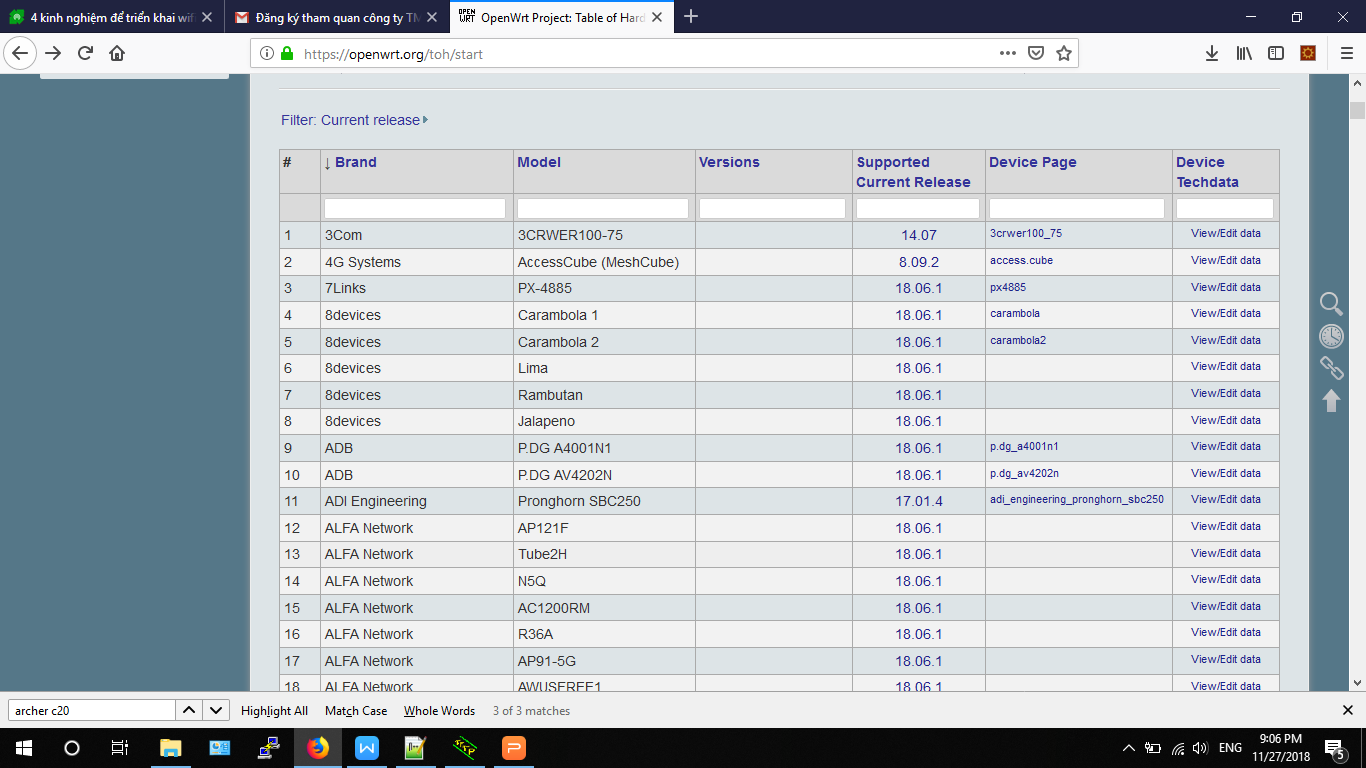
### Lựa chọn thiết bị

Đối với các thiết bị router Wi-Fi, muốn cài đặt được firmware từ OpenWrt thì phải đáp ứng được các yếu tố:

* Hỗ trợ Linux
* Có trong danh sách các thiết bị được hỗ trợ bởi OpenWrt

Với yếu tố là phải có trong danh sách các thiết bị được hỗ trợ bởi OpenWrt thì theo đó, một router Wi-Fi muốn cài đặt được phải có tối thiểu 4MB bộ nhớ flash và 32MB bộ nhớ RAM. Tuy nhiên đó, đó chỉ là yêu cầu tối thiểu để cài được firmware OpenWrt, nếu muốn sử dụng OpenWrt một cách thoải mái thì cần có 8MB bộ nhớ flash và 64MB bộ nhớ RAM. Vì yêu cầu phần cứng khá cao để có thể sử dụng được các gói và các ứng dụng khác nên chúng tôi mới tiến hành nghiên cứu thị trường router Wi-Fi ở Việt Nam và chúng tôi đề xuất chọn router AC750-Archer C20 version 4 được cung cấp bởi hãng Tp-Link có thể đáp ứng được các yếu trên đồng thời nó có mức giá khá hợp lý. Không dừng lại ở đó, chúng tôi sử dụng thêm 2 router Wi-Fi phụ để thử nghiệm trong luận văn này.

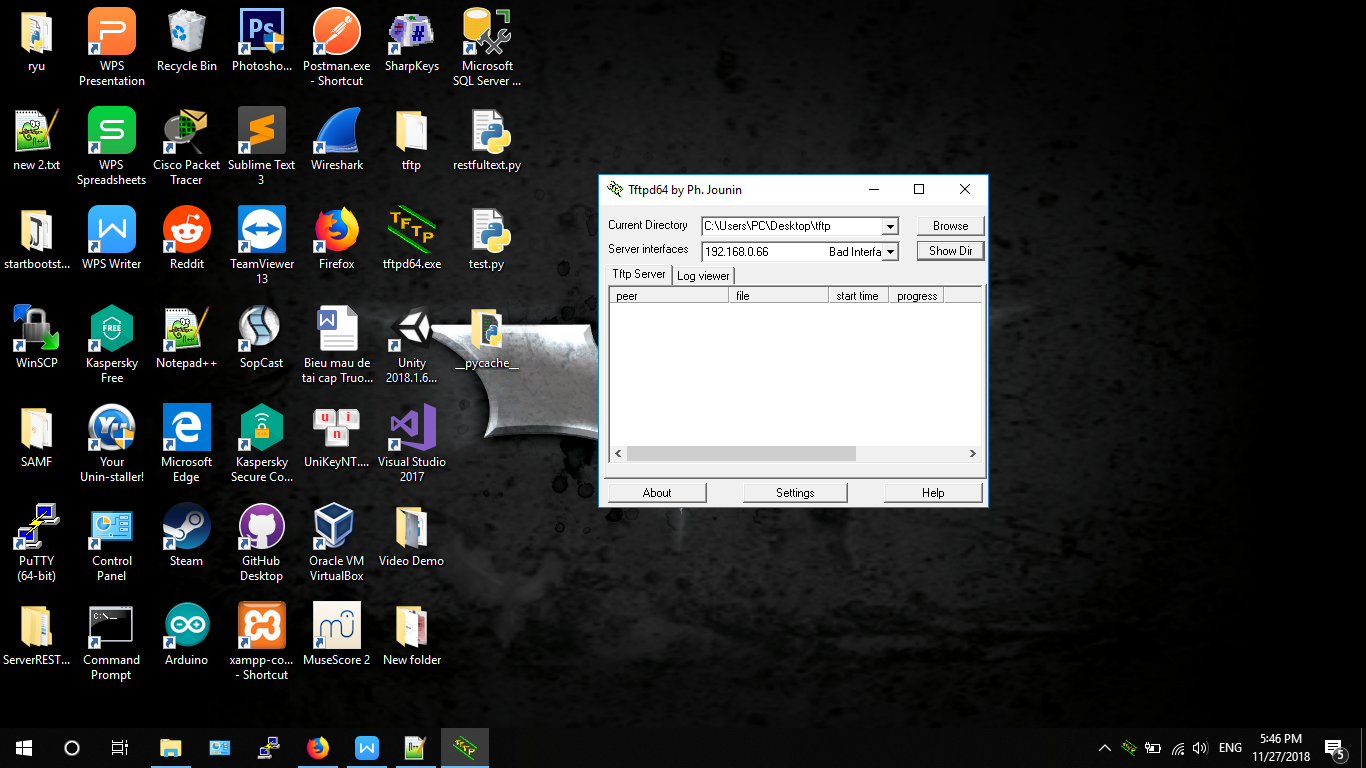
Mỗi router Wi-Fi có phần cứng khác nhau thì phải lựa chọn firmware do OpenWrt cung cấp đúng với phần cứng của mình. Để tìm được firmware phù hợp, ta cần phải vào website <https://openwrt.org/toh/start> để tìm và tải firmware về máy tính để chuẩn bị cho bước cài đặt như hình 3.1



1. Danh sách router được OpenWrt hỗ trợ

### Cài đặt firmware cho router

Khi mua mới một router Wi-Fi từ nhà cung cấp, thường ta có thể tiến hành đổi firmware khác bằng cách sử dụng giao diện website mặc định của router đó. Tuy nhiên có rất nhiều router không cho phép thay đổi firmware bằng website để tránh tình trạng người dùng không có kinh nghiệm và hiểu biết làm hỏng đi firmware. Do đó chúng tôi mới sử dụng công cụ thay thế là phần mềm Tftpd64 để tạo một TFTP server và có thể dùng chung cho mọi router để thay đổi firmware.



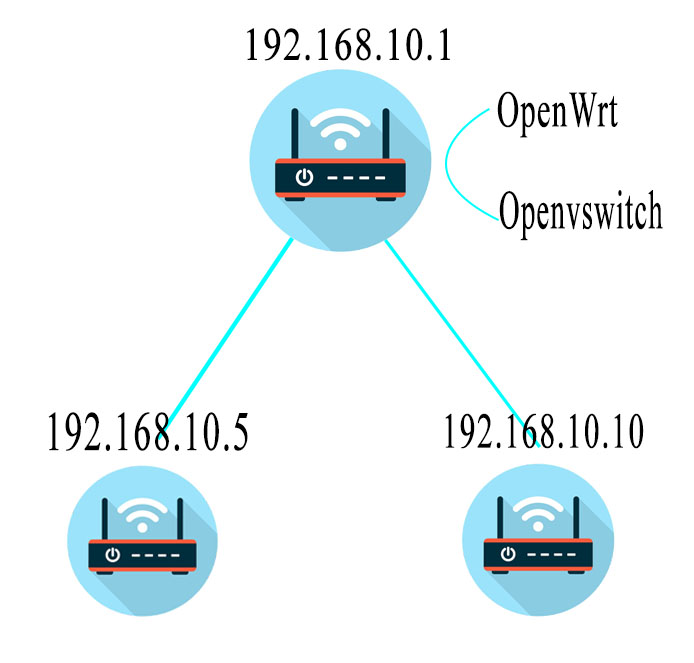
1. Phần mềm TFTPD64

Quy trình cài đặt firmware từ TFTP server làm theo các bước sau:

* Bước 1: Khởi động phần mềm tftpd64
* Bước 2: Chọn Browse để tìm đến thư mục chứa firmware trên máy tính.
* Bước 3: Đặt IP tĩnh cho máy chủ. Lưu ý là địa chỉ IP này phải cùng nhánh mạng với IP router khi chưa đổi firmware và phải khác địa chỉ IP của router.
* Bước 4: Cắm cáp RJ45 nối vào cổng LAN trên máy tính và cổng LAN trên router
* Bước 5: Khởi động lại router và nhấn nút WPS/Reset trên router khi đang trong quá trình khởi động. Router sẽ bắt đầu tìm và nhận firmware về.
* Bước 6: Đợi đến khi tiến trình truyền tập tin kết thúc thì có thể đóng server lại.

### Lựa chọn hình trạng mạng

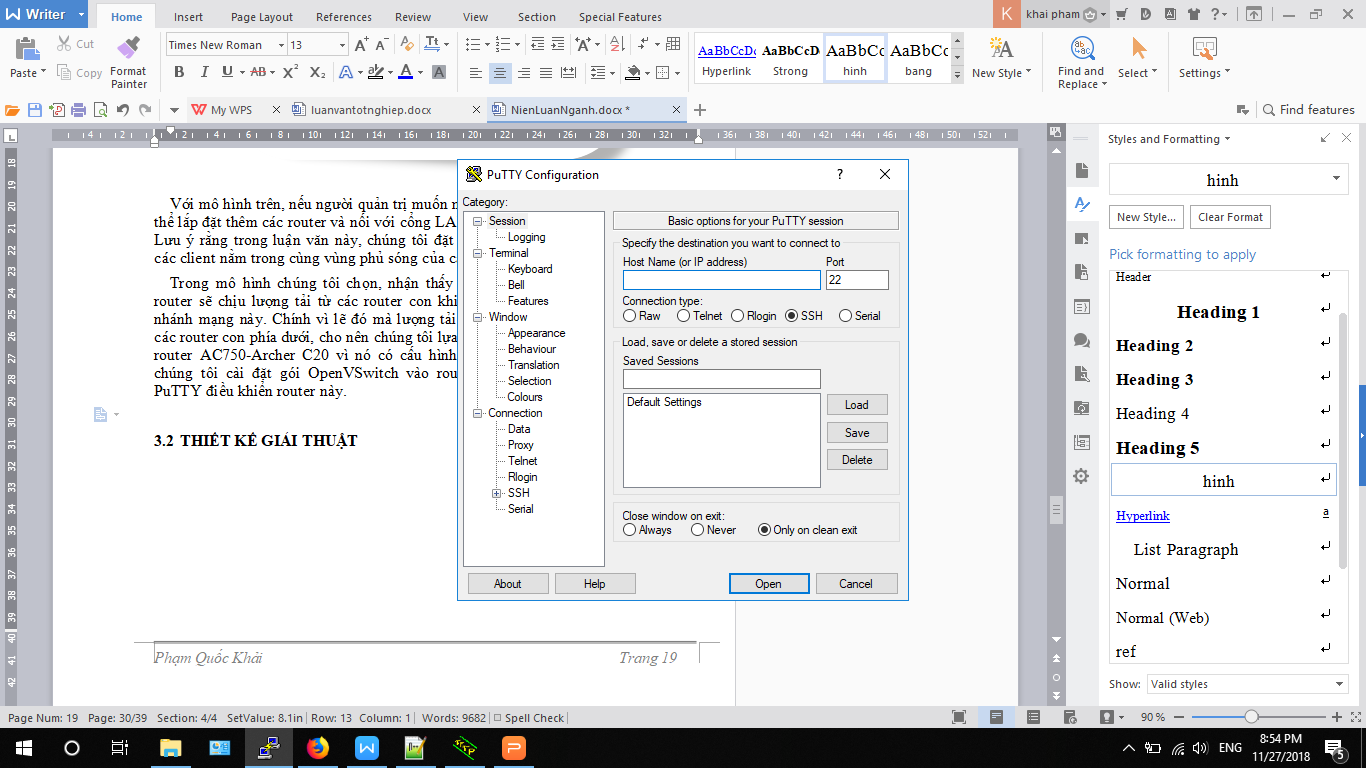
Chúng tôi cân nhắc lựa chọn một hình trạng mạng phổ biến đối với mạng Wi-Fi và quyết định chọn một router Wi-Fi có cấu hình tốt để làm gateway cho nhánh mạng và mở rộng bằng 2 router có cấu hình trung bình như hình 3.3



1. Mô hình mạng trong nghiên cứu

Với mô hình trên, nếu người quản trị muốn mở rộng Wi-Fi cho nhánh mạng thì có thể lắp đặt thêm các router và nối với cổng LAN của router gateway hoặc router phụ. Lưu ý rằng trong luận văn này, chúng tôi đặt 3 router Wi-Fi trên gần nhau và cho các client nằm trong cùng vùng phủ sóng của cả 3 router.

Trong mô hình chúng tôi chọn, nhận thấy rằng router đóng vai trò gateway là router sẽ chịu lượng tải từ các router con khi có các gói tin có đích đến là ngoài nhánh mạng này. Chính vì lẽ đó mà lượng tải của gateway nhận sẽ lớn hơn so với các router con phía dưới, cho nên chúng tôi lựa chọn giải pháp là cho gateway này là router AC750-Archer C20 vì nó có cấu hình mạnh hơn cách router con. Sau đó chúng tôi cài đặt gói OpenVSwitch vào router này bằng cách dùng phần mềm PuTTY điều khiển router này.



1. Giao diện phần mềm PuTTY

Để điều khiển được router, ta cần phải nhập địa chỉ IP vào ô Host Name (or IP address) của PuTTY và đăng nhập vai trò người dùng là root. Sau đó, ta tiến hành sử dụng dòng lệnh để tải gói cài đặt OpenVSwitch về bằng lệnh

#opkg update

#opkg install openvswitch

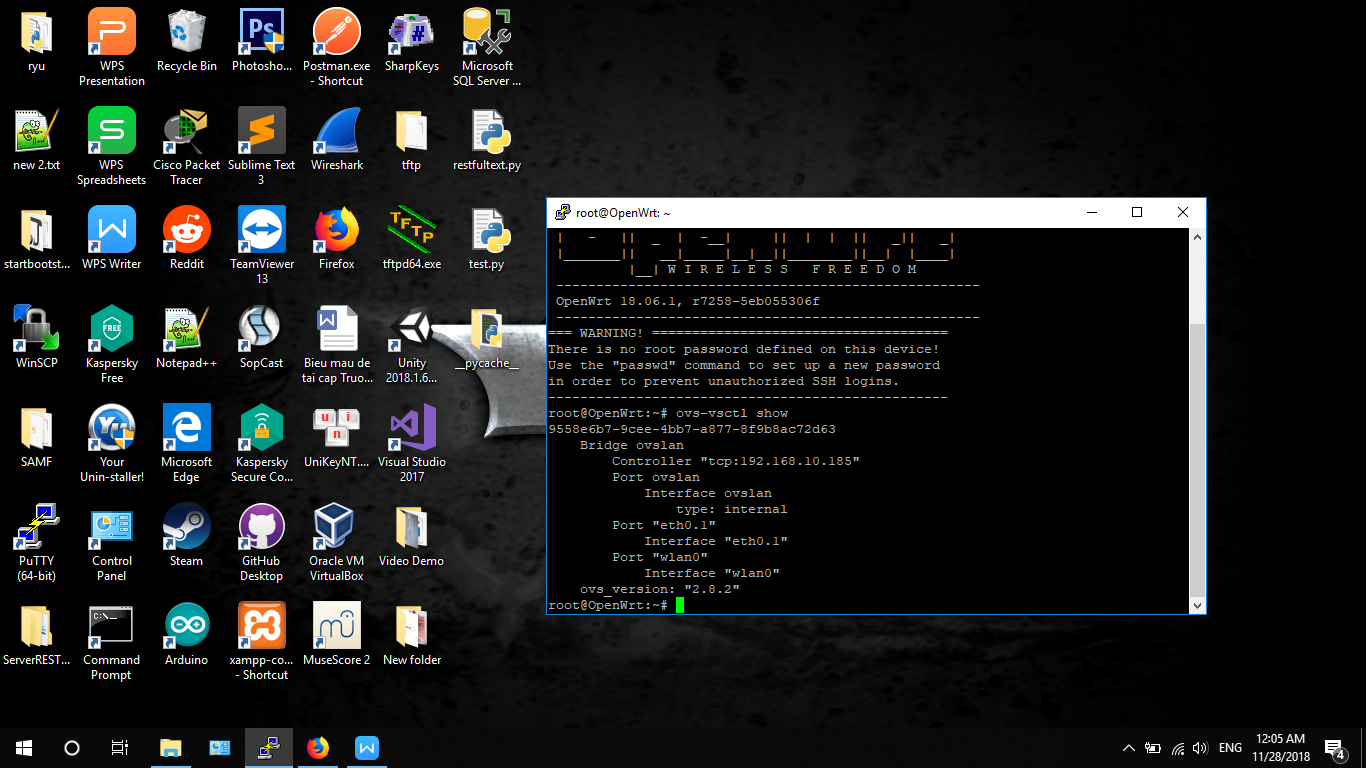
Sau khi đã cài xong gói OpenVSwitch, ta phải tạo một cầu nối bằng OpenVSwitch nối đến các giao diện mạng của router để nhờ bộ điều khiển giám sát và điều khiển luồng dữ liệu thông qua cầu nối này. Để tạo cầu nối bằng OpenVSwitch ta sử dụng lệnh sau:

#ovs-vsctl add-br <BRIDGE-NAME>

#ovs-vsctl add-port <BRIGDE-NAME> <PORT-NAME>

Câu lệnh thứ 2 dùng để gắn giao diện mạng của OpenWrt vào với cầu nối của OpenVSwitch. Tuy nhiên việc thực hiện gắn kết cầu nối và giao diện mạng không thể làm bằng tay bởi vì khi cấu hình bằng tay thì xảy ra lỗi không thể truy cập router nữa. Chính vì thế, chúng tôi khuyến cáo nên sử dụng một tập tin chứa một kịch bản và chạy nó để không bị lỗi, sau đó khởi động lại router. Sau khi đã nối cầu nối OpenVSwitch với giao diện mạng của OpenWrt xong, ta sử dụng lệnh sau để cấu hình cho cầu nối được phép điều khiển bởi controller:

#ovs-vsctl set-controller <BRIDGE-NAME> tcp:<IP-CONTROLLER>:<PORT>



1. Các giao diện nối với cầu nối của OpenVSwitch

Việc cài đặt OpenVSwitch và cho phép controller điều khiển luồng dữ liệu nhằm mục đích tách chức năng chuyển mạch trong router sang controller, điều này làm giảm lượng tải phải xử lý đối với router gateway.

Trên mỗi router, chúng tôi cài đặt gói luci-mod-rpc bằng cách sử dụng dòng lệnh:

#opkg install luci-mod-rpc

Gói này nhằm giúp cho việc truy vấn và điều khiển router từ xa mà không cần phải sử dụng SSH.

### Xây dựng bộ điều khiển

Trong luận văn này, chúng tôi hướng đến việc tạo một bộ điều khiển có thể làm được 2 công việc:

* Thứ nhất là điều khiển luồng dữ liệu trên router gateway
* Thứ hai là chạy giải thuật cân bằng tải cho hệ thống Wi-Fi.

Để làm được điều này, chúng tôi quyết định sử dụng Ryu framework để chạy trên bộ điều khiển. Một chương trình tên Simple\_switch sẽ giúp điều khiển luồng dữ liệu trên router đã có cài đặt OpenVSwitch và cho phép controller điều khiển. Bên cạnh đó, một chương trình viết bằng ngôn ngữ python để chạy giải thuật cân bằng tải cũng sẽ được chạy trên chính bộ điều khiển này.

### Thiết kế giải thuật cân bằng tải

Như đã đề cập ở mục 3.13, chúng tôi cho các client vào vùng giao nhau của các router. Chính nhờ có điều kiện đó, chúng tôi lợi dụng chức năng roaming trong mạng Wi-Fi để thực hiện phân bố lại các client đang kết nối trong hệ thống. Cụ thể như sau:

Đầu tiên, các router con phải vô hiệu hóa chương trình dnsmasq của OpenWrt để client có thể roaming sang các router khác bằng cách thực hiện câu lệnh sau:

#/etc/init.d/dnsmasq disable

Tiếp theo, chúng tôi định nghĩa mức tải đối với router là mức sử dụng CPU ở thời điểm hiện tại và mức quá tải là mức CPU cao nhất mà router có thể hoạt động bình thường. Thêm nữa, chúng tôi cũng định nghĩa mức cảnh báo là mức CPU mà router có khả năng quá tải nếu có thêm nối kết đến nó.

Trên bộ điều khiển chúng tôi quyết định định kỳ truy vấn mức sử dụng CPU của các router trong hệ thống. Trong trường hợp có một router bất kỳ bị quá tải, bộ điều khiển sẽ chạy giải thuật cân bằng tải cho hệ thống. Tiến trình cần bằng tải diễn ra như sau:

Bước 1: Bộ điều khiển truy vấn danh sách các client đang kết nối đến router bị quá tải.

Bước 2: Tìm kiếm các router đang có mức CPU dưới mức cảnh báo và cùng nhóm với router bị quá tải

Bước 3: Bộ điều khiển yêu cầu loại bỏ 1 client khỏi router bị quá tải và chỉ cho client đó roaming đến các router tìm được ở bước 2.

### Xây dựng website theo dõi hệ thống

Để quản trị và theo dõi thông tin của hệ thống mạng một cách toàn diện hơn chúng tôi xây dựng và thiết kế một website quản trị có khả năng cung cấp thông tin, theo dõi các phiên nối kết, đưa ra các cảnh báo mạng và có thể cấu hình cho hệ thống từ phía server.

Ngôn ngữ chính được chúng tôi sử dụng cho phần này là php. Vì php có khả năng nối kết cơ sở dữ liệu, được hỗ trợ bởi ajax, có thể thực hiện các công việc quan trọng ở cả phía client,server và quan trọng nó có thể sử dụng để truy vấn resful api từ firmware openwrt nên sẽ là một ngôn ngữ-công cụ hiệu quả nhất có thể sử dụng.

Các bước thực hiện truy vấn thông tin từ website:

* Lập trình các tập lệnh command line cùng một số shell script cần thiết trên hệ thống các router để đảm bảo có thể truy vấn mọi thông tin của hệ thống mà resful api cung cấp bởi openwrt không thể truy vấn được.
* Code php được sử dụng để gửi yêu cầu restful kèm theo các chuỗi json đến các router thông qua ip của chúng để truy vấn thông tin từ router, ra lệnh cấu hình cho router hoặc thực hiện các shell script đã tạo ra trước đó.
* Các dữ liệu được trả về sau khi gửi thông điệp Restful là các chuỗi json thô chưa được xử lý bao gồm nhiều thông tin dư thừa khác ngoài thông tin mong muốn nên chúng sẽ được cắt tỉa hợp lý để phù hợp với yêu cầu của trang web.
* Đối với các thông tin cần trả về theo thời gian thực như mức sử dụng CPU thì không thể sử dụng theo cách gửi Restful thông thường. Nên chúng sẽ được sử lý bằng một file ajax yêu cầu gửi đình kỳ các yêu cầu Restful đến router theo một chu kỳ mỗi 5 giây một lần
* Một trong những chức năng chính của website là tương tác với các router từ server. Để thực hiện được việc này nhóm sử dụng thư viện UCI của OpenWrt. UCI chịu trách nhiệm cho việc cấu hình router bằng dòng lệnh mà không cần can thiệp vào các file cấu hình tạo sự linh động dễ dàng trong việc cấu hình các router từ xa.
* Ngoài ra đây mục đích của website là theo dõi được tất cả các router trong hệ thống mạng. Nên để tối ưu hóa việc quản lý và truy vấn thông tin của các router được dễ dàng hơn chúng tôi tạo ra một cơ sở dữ liệu lưu trữ các thông tin cơ bản của router. Thông tin các router này sẽ được cập nhật bởi người quản trị website tùy theo sự thay đổi của hạ tầng mạng.

Sơ đồ hoạt động của hệ thống được biểu thị như sau:

Cơ sở dữ liệu

Giao diện quản trị

yêu cầu thực hiện

Chuỗi json thô

Yêu cầu restful

Yêu cầu truy vấn

Thông tin hệ thống

Thông tin đã xử lý

Thông tin hệ thống

Bộ xử lý PHP

ajax

Hệ thống router

1. Luồng tương tác của website

**Xây dựng các chức năng chính của website**

Cơ bản website sẽ xây dựng các hàm chức năng theo sơ đồ cây chức năng sau:

Bật tắt chế độ wireless

Trang quản lý chung

Khởi động lại router

Xem thông tin router

Quản lý nối kết

Tính toán đưa ra cảnh báo

Cập nhập tình trạng cpu

Truy cập router hệ thống

1. Chức năng của website

Để thực hiện theo sơ đồ đã đình ra các hàm chức năng sẽ chia làm ba nhóm và thực hiện bằng các bước khác nhau:

* **Các hàm truy vấn thông tin thông thường.**

Như đã trình bày ở phần ý tưởng để xây dựng các hàm này nhóm thực hiện bằng 5 bước:

* Xây dựng tập lệnh command line để lấy thông tin từ router
* Xây dựng hàm lấy token để sử dụng được API mà OpenWrt cung cấp
* Sử dụng token để gửi chuỗi Restful yêu cầu router thực hiện các command đã xây dựng
* Xử lý chuỗi trả về lược bỏ các thông tin dư thừa
* Kết nối các thông tin đã xử lý với các thẻ html bằng javascript
* **Các hàm truy vấn thông tin theo thời gian thực**

Để xây dựng những hàm này các bước thực hiện sẽ giống với xây dựng các hàm truy vấn thông tin thông thường. Tuy nhiên sau khi đã hiển thị được thông tin ra trang web nhóm xây dựng thêm một hàm ajax cho phép gửi các thông điệp Restful tuần tự mỗi 5 giây một lần.

* **Các hàm cấu hình cho router**

Như giới thiệu ở đầu phần này sẽ sử dụng tập lệnh UCI để cấu hình cho các router. Cách thức hoạt động của tập lệnh uci có thể mô tả như sau:

* Mọi thông tin cấu hình khi sử dụng uci sẽ tạm thời được lưu lại vào một tập tin cấu hình tạm và không được ghi vào tập tin cấu hình chính thức.
* Có thể xem lại các thay đổi cấu hình này trước khi thực hiện việc thay đổi cấu hình của router.
* Thực hiện lệnh commit của uci để xác nhận việc thực hiện tập tin cấu hình.
* Khởi động lại dịch vụ mạng để hoàn tất phần cấu hình.

Như vậy để xây dựng chức năng cấu hình cho webiste cần thực hiện theo các bước sau:

* Xây dựng tập lênh uci cấu hình cho router.
* Xây dưng hàm lấy token để sử dụng được API mà OpenWrt cung cấp.
* Sử dụng token để gửi chuỗi Restful yêu cầu router thực hiện tập lệnh uci đã xây dựng.
* Gửi chuỗi restful yêu cầu router khởi động lại dịch vụ mạng.
* **Kết nối các router trên một giao diện chung**

Sau khi xây dựng riêng các chức năng riêng cho từng router công việc tiếp theo là kết nối lại các router và thông tin của chúng trên cùng một giao diện.

* Đầu tiên website kết nối đến cơ sở dữ liệu và truy vấn các thông tin cần thiết.
* Sau khi có được số lượng router trên hệ thống cùng các ip liên quan website sẽ gửi yêu cầu restful đến tất cả các router này.
* Nhờ các hàm và các tập lệnh uci,shell script được xây dựng trước quá trình truy vấn được thực hiện và trả về toàn bộ thông tin của hệ thống.
* Các thông số lấy được sẽ được website tính toán và xử lý để đưa ra những cảnh báo thích hợp.
* Thông số tải trung bình cũng như số lượng nối kết cũng được website xử lý và hiển thị trên giao diện.

## 3.2 KẾT QUẢ NGHIÊN CỨU

### 3.2.1 Hệ thống topology openwrt

Sau thời gian nghiên cứu nhóm đã hoàn thành xong topology gồm 3 router được cài đặt firmware openwrt trong đó có:

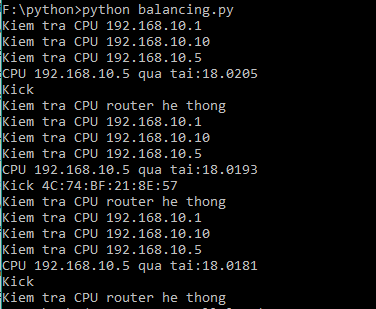
* Một router chính tp-link acher c20 có cài cả firmware openwrt và openvswitch có khả năng kết nối với controller để nhận các lệnh điều khiển dòng dữ liệu và điều khiển nối kết.
* Hai router con tplink-741 và tplink-841 được cài firmware openwrt có khả năng chia sẽ dữ liệu và nhận điều khiển từ controller.
* Mô hình topology thực tế như sau:



1. Hình trạng mạng thực tế

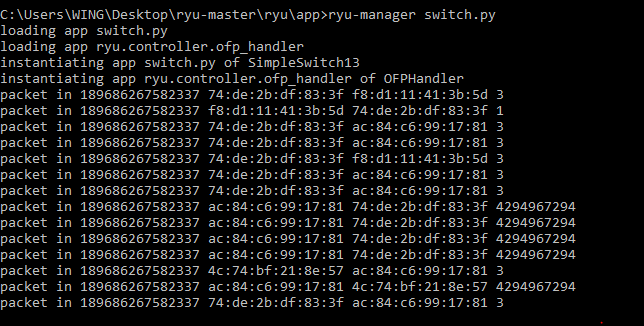
### 3.2.2 Controller điều khiển hệ thống

* Controller nhận các thông tin từ router tính toán đưa ra các xử lý điều khiển nối kết dựa trên công nghệ Restful đã hoàn thành các chức năng cơ bản.Nó có thể chuyển nối kết các client nếu thông số CPU trên mức ngưỡng giúp giảm tải cho router hiện tại và chia cho các router cho hệ thống đang có ít nối kết. Ngoài ra controller còn có thể hiện cảnh báo khi phát hiện router không thể gửi dữ liệu đến được-báo hiệu các vấn đề về dây hoặc đường truyền mạng



1. Màn hình xử lý cân bằng tải

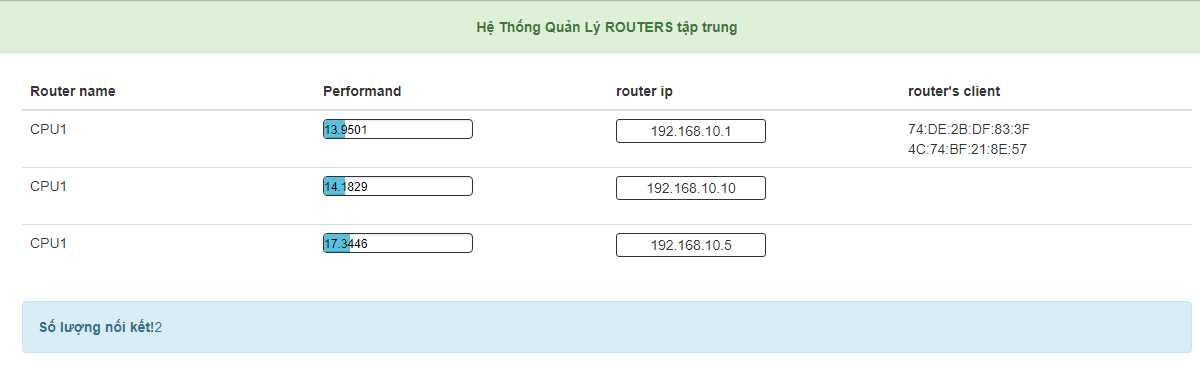
* Phần controller điều khiển dữ liệu bằng giao thức openflow trên framework ryu cũng được áp dụng thành công.



1. Điều khiển luồng dữ liệu trên gateway

### 3.2.3 Website quản trị

Như mô hình đã đề ra giao diện chính của website quản trị hệ thống quản lý tất các thông tin cơ bản của hệ thống.



1. Màn hình quản lý chung trên website

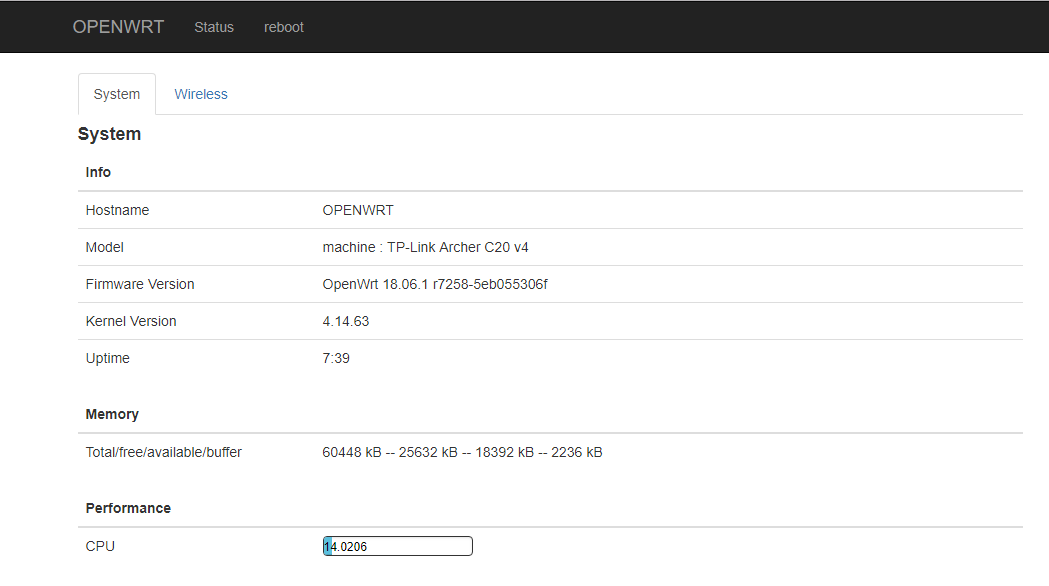
Khi các thông số mà website nhận được vượt qua ngưỡng nó có thể đưa ra các cảnh báo cho người quản trị.

(Vì các router đang trong tình trạng ổn định nên thông số ngưỡng cpu được đặt ở mức 15% để kiểm tra code)

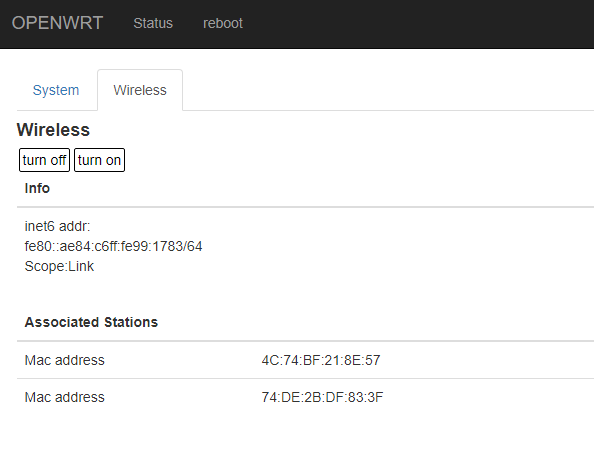


1. Cảnh báo quá tải

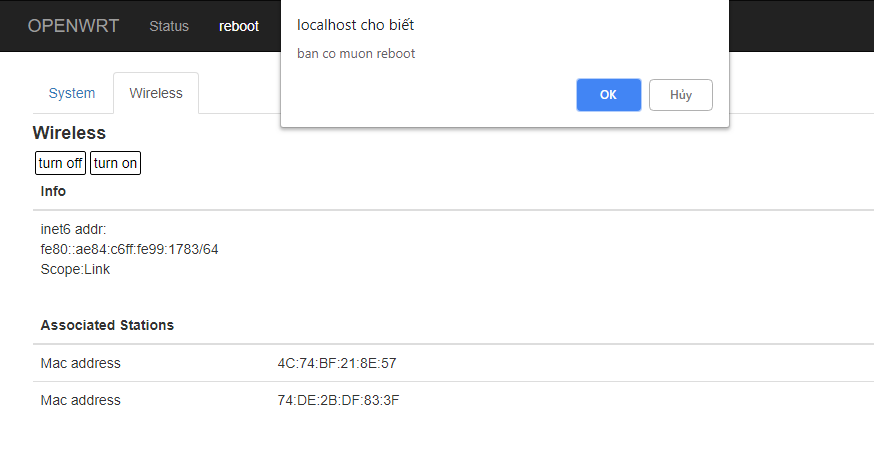
Click vào các địa chỉ ip thông tin chi tiết các các router cùng các nối kết sẽ hiện ra cùng các chức năng phụ như khởi động lại router bật,tắt wifi.



1. Thông tin cơ bản trên router



1. Thông tin giao diện Wi-Fi của router



1. Chức năng reboot

# 

# CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## KẾT LUẬN

Sau khoảng thời gian nghiên cứu, tìm tòi và học hỏi, nhóm đã hoàn thành nghiên cứu và tìm hiểu được nhiều kiến thức mới:

- Tìm hiểu cơ bản về cách xây dựng topology thực tế.

- Tìm hiểu router, nguyên lý hoạt động và cách cấu hình của chúng.

- Tìm hiểu được thêm nhiều firmware và công nghệ mới(openwrt, openvswitch, openflow...).

- Áp dụng được các kiến thức đã học kết hợp chúng để tạo ra hệ thống hoàn chỉnh.

Nhóm đã giải quyết cơ bản vấn đề giảm tải cho wifi trong hệ thống mạng và hoàn thành các mục tiêu đề ra tuy nhiên bên cạnh đó đề tài vẫn còn những điểm hạn chế cần khắc phục:

- Việc cấu hình router cần phải khởi động lại dịch vụ mạng nên có thể làm gián đoạn cả controller và CSDL của website.

- Chưa kết hợp được controller điều khiển nối kết và controller điều khiển dòng dữ liệu.

- Giải thuật điều khiển tải chưa hoản toàn tối ưu có thể kết hợp thông tin về cường độ tín hiệu nối kết của các thiết bị để giúp giải thuật chính xác hơn.

## HƯỚNG PHÁT TRIỂN

Đề tài bước đầu chỉ là tìm hiểu và thực xây dựng hệ thống mạng qui mô nhỏ, hướng đến mục tiêu sẽ áp dụng đến tạo ra các hệ thống lớn với giải thuật điều khiển nhanh và chính xác hơn.

Thêm các module cho controller để thực hiện các công việc như cấu hình tự động và gán kênh tự động cho hệ thống nhằm tăng thông lượng cho hệ thống một cách triệt để

Kết hợp được controller điều khiển dữ liệu và điều khiển nối kết làm cho bộ điều khiển dễ dàng sử dụng hơn.

# PHỤ LỤC

# 

# TÀI LIỆU THAM KHẢO

1. **Ying-Dar Lin.** *SAMF: An SDN-Based Framework for Access Point Management in Large-scale Wi-Fi Networks.* [National Chiao Tung University - 2017](https://www.nctu.edu.tw/en)
2. [www.pcworld.com](https://www.nctu.edu.tw/en)
3. [https://github.com/OpenWrt/luci/wiki/JsonRpcHowTo](https://www.nctu.edu.tw/en)
4. [https://openwrt.org/docs/guide-user/base-system/uci](https://www.nctu.edu.tw/en)
5. [https://w1.fi/hostapd/](https://www.nctu.edu.tw/en)
6. https://bachkhoa-aptech.edu.vn
7. https://osrg.github.io/ryu/