BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CẦN THƠ

KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

NGÀNH CÔNG NGHỆ THÔNG TIN

Đề tài

HỆ THỐNG QUẢN LÝ HIỆU NĂNG CỦA CÁC ACCESS POINT TRONG MẠNG WIFI CỠ LỚN SỬ DỤNG BỘ ĐIỀU KHIỂN TẬP TRUNG

Phân hệ:Phân hệ cân bằng tải cho các access point

Sinh viên thực hiện:Phạm Quốc Khải

Mã số: B1401149

Khóa:40

Cần Thơ, 12/2018

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CẦN THƠ

KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BỘ MÔN CÔNG NGHỆ THÔNG TIN

LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

NGÀNH CÔNG NGHỆ THÔNG TIN

Đề tài

HỆ THỐNG QUẢN LÝ HIỆU NĂNG CỦA CÁC ACCESS POINT TRONG MẠNG WIFI CỠ LỚN SỬ DỤNG BỘ ĐIỀU KHIỂN TẬP TRUNG

Phân hệ:Phân hệ cân bằng tải cho các access point

Giáo viên hướng dẫn

TS.Thái Minh Tuấn

Sinh viên thực hiện

Phạm Quốc Khải

Mã số: B1401149

Khóa:40

Cần Thơ, 12/2018

NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

…….Ngày…...tháng…..năm 2018

Giảng viên

(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, tôi xin tỏ lòng biết ơn đến gia đình đã quan tâm, ủng hộ về vật chất lẫn tinh thần trong quá trình 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 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 tôi thực hiện luận văn. Cuối cùng tôi xin cảm ơn bạn bè, anh chị khoa Công nghệ thông tin và Truyền thông đã cho tôi ý tưởng, hỗ trợ, động viên 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 tôi mong nhận được những góp ý của thầy cô và các bạn.

Phạm Quốc Khải

MỤC LỤC

DANH SÁCH CÁC TỪ VIẾT TẮT

STT	Từ viết tắt	Từ viết đầy đủ
1	AP	Access Point
2	API	Application Programming Interface
3	CSDL	Cơ sở dữ liệu
4	MIMO	Multiple-Input Multiple-Output
5	RPC	Remote Procedure Calls
6	SDN	Software Define Networking
7	SSH	Secure Socket Shell
8	UCI	User Config Interface
9	Wi-Fi	Wiless Fidelity

DANH SÁCH HÌNH

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.

Để giải quyết vấn đề quá tải nói trên, đã có nhiều hệ thống cân bằng tải cho Wi-Fi được nghiên cứu và xây dựng nhưng phải đầu tư với chi phí khá cao nên ở luận văn này,chúng 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, SDN.

ABSTRACT

Wi-Fi is becoming more common for 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, SDN*

CHƯƠNG 1: TỔNG QUAN

1.1 ĐỘNG CƠ NGHIÊN CỨU

Wi-Fi đã trở thành công nghệ 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(Cisco Meraki,Aerohive,Unifi) dành riêng cho các điểm truy cập Wi-Fi quy mô lớn tuy nhiên thông thường giá của các router này khá cao và công nghệ tạo ra chúng không được công khai Đồng thời phải thuê các controller điều khiển.

Vì các lý do kể trên nhóm quyết định thực hiện đề tài "HỆ THỐNG QUẢN LÝ HIỆU NĂNG CỦA CÁC ACCESS POINT TRONG MẠNG WIFI CỠ LỚN SỬ DỤNG BỘ ĐIỀU KHIỂN TẬP TRUNG" đề tài này sử

dụng các tiện ích trên firmware OpenWrt cũng như xây dựng 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.

1.2 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.

1.3 NHỮNG NGHIÊN CỨU LIÊN QUAN

Bài báo "SAMF: An SDN-Based Framework for Access Point Management in Large-scale Wi-Fi Networks"- Ying-Dar Lin, Fellow, IEEE, Chun-Hung Hsu, Minh-Tuan Thai, Chien-Ting Wang, Yi-Jen Lu, and Yi-Ta Chiang [1]

Bài báo này trình bài về framework SAMF-một framework mã nguồn mở có thể lập trình được và phổ biến cho việc quản trị ap trong mạng wifi lớn Việc chấp sử dụng công nghệ SDN và giao thức OpenFlow, SAMF có thể nhanh chóng được triển khai trên những AP chi phí phần cứng thấp và trên Controller đám mây, đồng thời cho phép những dịch vụ mạng mới được tích hợp nhanh chóng.Kết quả đạt được của công nghệ SAMF là tăng thông lượng hệ thống đến 26,5% và cải thiện độ cân bằng của hệ thống khoảng 40%.

Ubiquiti Network Management System

Ubiquiti nhận thấy rằng hầu hết các AP đều có thể cấu hình trực tiếp trên nền tảng web cho phép cấu hình từng thiết bị AP riêng lẽ. Tuy nhiên

đối với một hệ thống lớn việc cấu hình này sẽ tốn nhiều thời gian và không hiệu quả. Do đó Ubiquiti (công ty công nghệ) đã đưa ra một giải pháp thông minh là sử dụng mộ bộ kiểm soát (controller) để quản lý tập chung dữ liệu kiểm soát thông tin từ các AP. Nhưng việc bổ sung bộ kiểm soát này bắt buộc phải kèm theo một thiết bị phần cứng do ubiquiti cung cấp để giao tiếp với các thiết bị AP và quản trị các AP này.[2]

1.4 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

2.1 MẠNG WI-FI

2.1.1 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 như toàn bộ các thiết bị điện tử ngày nay như điện thoại di động, laptop… đều có trang bị công nghệ để có thể kết nối Wi-Fi[3]

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

2.1.2 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..[4]

2.1.3 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.

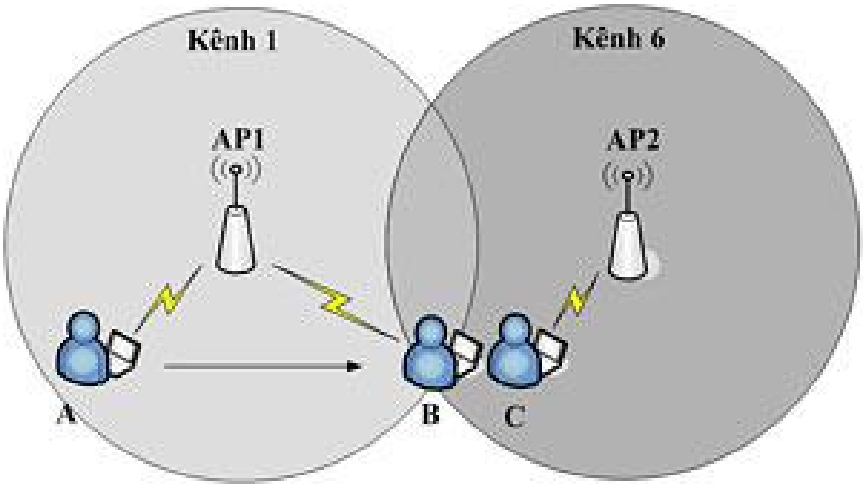
Trình điều khiển card Wi-Fi của client sẽ quyết định lúc nào cần thực hiện roaming. Giải thuật roaming là các giải thuật bí mật của nhà sản xuất card Wi-Fi, ta không thể nhìn thấy thời điểm chính xác khi nào diễn ra tiến trình roaming. Giải thuật roaming thường căn cứ vào cường độ tín hiệu, chất lượng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



Hình 2.1. 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[i.] [5]

2.2 FIRMWARE OPENWRT

2.2.1 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). Có khoảng 3.500 gói phần mềm tùy chọn có sẵn để cài đặt qua hệ thống quản lý gói opkg.

2.2.2 Những tính năng chính 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, WNIC s, modem DSL s, FX, các nút phần cứng có sẵn, vv

Luci-giao diện website quản trị được phát triển cho openwrt. 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. [6]

2.3 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 hệ điều hành cho router để chạy theo tiến trình sau:

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

Trong trường hợp không tìm thấy hệ điều hành hoặc hệ điều hành 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 hệ điều hành 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. [7]

2.4 GIỚI THIỆU CÔNG NGHỆ SDN

2.4.1 SDN là gì?

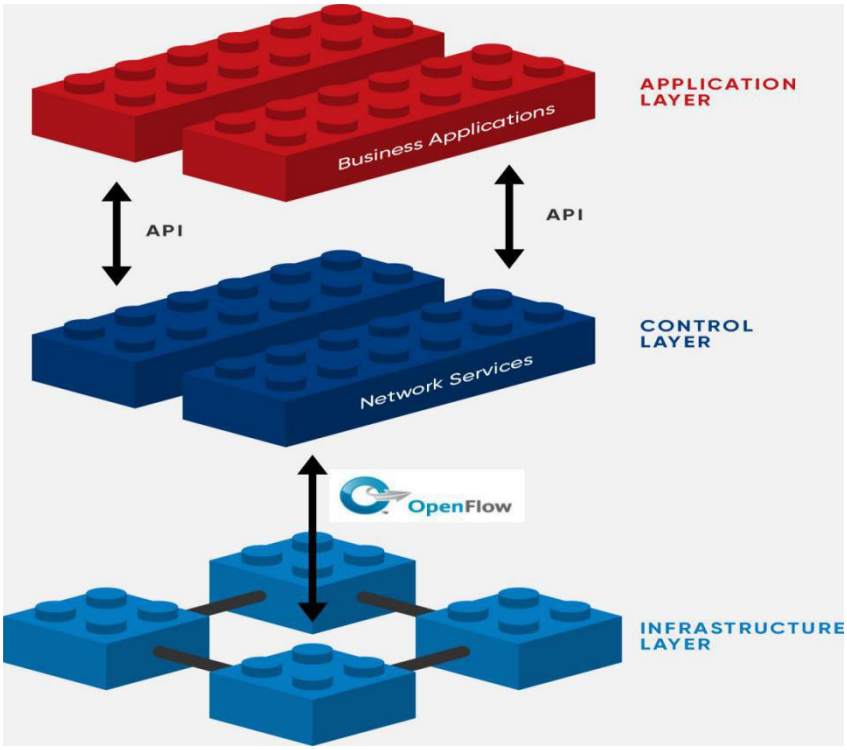
Mạng điều khiển bằng phần mềm (SDN) là một kiến trúc mới nổi có tính năng động, dễ quản lý, hiệu quả về chi phí và có thể thích nghi, làm cho nó trở nên lý tưởng việc xây dựng các ứng dụng mạng có tính chất linh động và có băng thông cao. Kiến trúc này tách các chức năng điều khiển và chuyển tiếp mạng 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 điều 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.[8]



Hình 2.2. Kiến trúc SDN

2.4.2 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.[9]

2.4.3 OpenVSwitch

Open vSwitch là phần mềm switch mã nguồn mở hỗ trợ giao thức OpenFlow. Open vSwitch được sử dụng với các hypervisors để kết nối giữa các máy ảo trên một host vật lý và các máy ảo giữa các host vật lý khác nhau qua mạng. Open vSwitch cũng được sử dụng trên một số thiết bị chuyển mạch vật lý. Open vSwitch là một trong những thành phần quan trọng hỗ trợ SDN (Software Defined Networking - Công nghệ mạng điều khiển bằng phần mềm)

OpenVSwitch 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.[10]

2.4.4 RYU Framework

Ryu là một phần mềm mã nguồn mở cung cấp các thư viện và công cụ cần thiết để phát triển các ứng dụng SDN. Ryu cung cấp một số thành phần cơ bản để kiểm soát và lập trình luồng dữ liệu cũng như hỗ trợ kiến trúc restful để tăng tính linh động trong việc truy vấn dữ liệu và cấu hình hệ thống. Lập trình viên có thể tự do tùy chỉnh các thành phần này cũng như

kết hợp chúng lại với nhau để tạo ra các thành phần mới cho ryu framwork. Các thành phần tạo nên ryu bao gồm:

- Thành phần thực thi chính.
- Thành phần điều khiển openflow.
- Thành phần giải mã và dịch mã.
- Thành phần ứng dụng.
- Thành phần điều khiển gói tin và cấu hình.

Ryu có chức năng điều khiển OpenFlow, vì vậy các ứng dụng SDN sử dụng công nghệ OpenFlow có thể dễ dàng được phát triển bằng cách sử dụng Ryu. Vì các phiên bản khác nhau của OpenFlow tồn tại,nên Ryu được thiết kế để tương thích với nhiều phiên bản và các bộ điều khiển OpenFlow khác nhau (bao gồm phiên bản 1.0, 1.2, 1.3 và 1.4).

Ngoài ra Ryu còn hỗ trợ các giao thức khác nhau để quản lý các thiết bị mạng, như OpenFlow, Netconf, OF-config. Tất cả các mã đều có sẵn miễn phí theo giấy phép Apache 2.0. [11]

2.5 Công nghệ Restful

Rest Là một kiểu 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 như: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.[12]

2.6 Thủ tục gọi hàm từ xa JSONRPC

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.

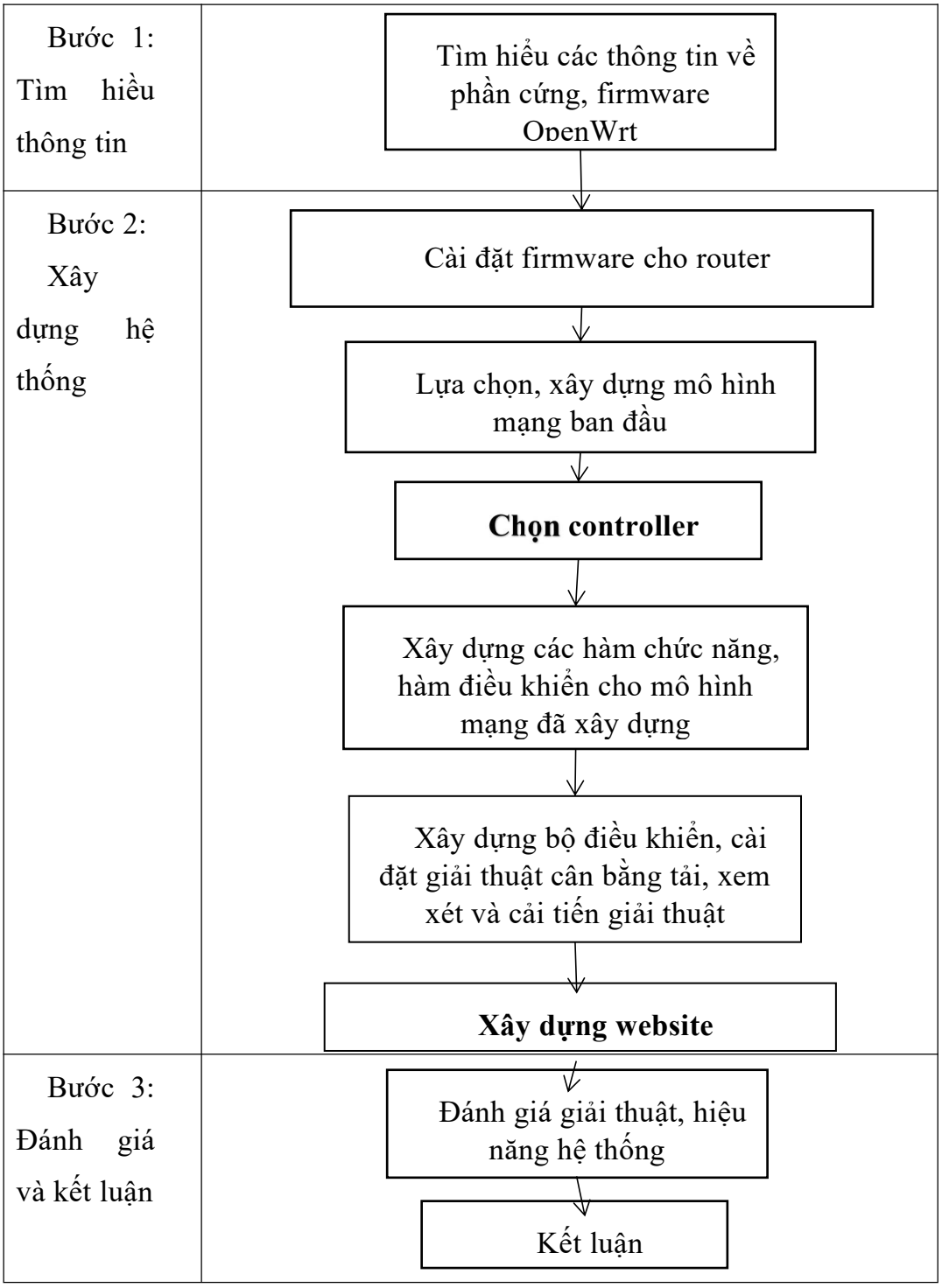
id - Id của yêu cầu mà nó đang phản hồi.

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. Tài liệu đầy đủ cho các lời gọi hàm này được cung cấp ở site: https://github.com/OpenWrt

CHƯƠNG 3: NỘI DUNG VÀ KẾT QUẢ NGHIÊN CỨU

3.1 NỘI DUNG NGHIÊN CỨU

Trong đề tài này, chúng tôi dựa theo sơ đồ sau để nghiên cứu và hoàn thiện sản phẩm:

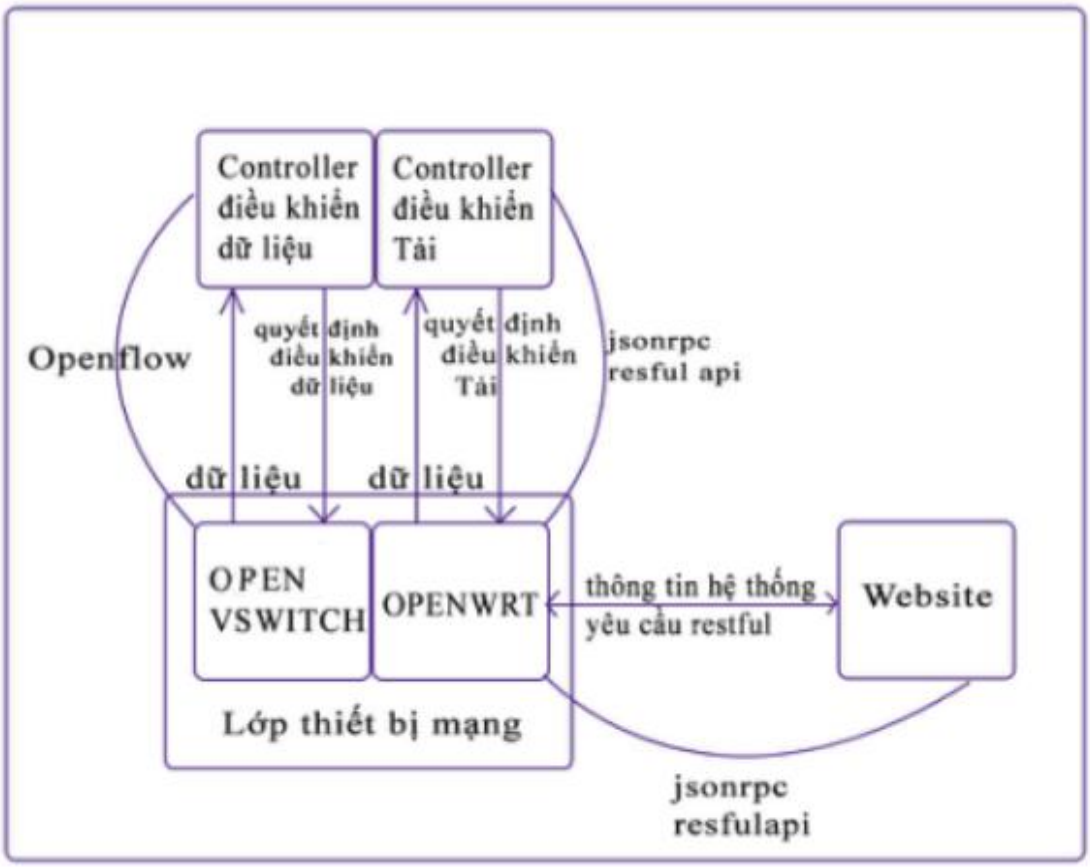


Cụ thể theo quy trình trên chúng tôi bắt đầu nghiên cứu bằng việc tìm hiểu những thiết bị router nào là phù hợp với đề tài bởi vì theo hướng dẫn từ website của OpenWrt (https://openwrt.org), không phải loại router nào trên thị trường đều có thể sử dụng firmware OpenWrt. Chính vì lí do đó, chúng tôi tiến 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.

3.1.1 Sơ đồ tổng quan hệ thống.



Hình 3.1. Sơ đồ tổng quan hệ thống

3.1.2 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.

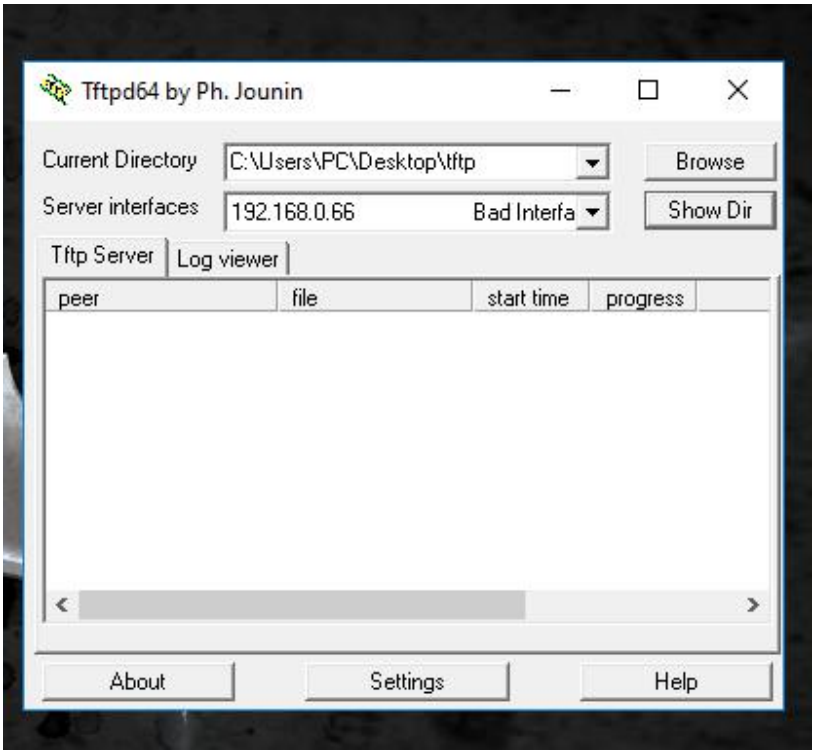
Mỗi router Wi-Fi có phần cứng khác nhau thì phải lựa chọn firmware do OpenWrt cung cấp đúng với phần cứng của mình. Để tìm được firmware phù hợp, ta cần phải vào website https://openwrt.org/toh/start để tìm và tải firmware về máy tính để chuẩn bị cho bước cài đặt như hình 3.2

↓ Brand	Model	Versions	Supported Current Release	Device Page	Device Techdata
3Com	3CRWER100-75		14.07	3crwer100_75	View/Edit data
4G Systems	AccessCube (MeshCube)		8.09.2	access.cube	View/Edit data
7Links	PX-4885		18.06.1	px4885	View/Edit data
8devices	Carambola 1		18.06.1	carambola	View/Edit data
8devices	Carambola 2		18.06.1	carambola2	View/Edit data
8devices	Lima		18.06.1		View/Edit data
8devices	Rambutan		18.06.1		View/Edit data
8devices	Jalapeno		18.06.1		View/Edit data
ADB	P.DG A4001N1		18.06.1	p.dg_a4001n1	View/Edit data
ADB	P.DG AV4202N		18.06.1	p.dg_av4202n	View/Edit data
ADI Engineering	Pronghorn SBC250		17.01.4	adi_engineering_pronghorn_sbc250	View/Edit data

Hình 3.2. Danh sách router được OpenWrt hỗ trợ

3.1.3 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.



Hình 3.3. 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.

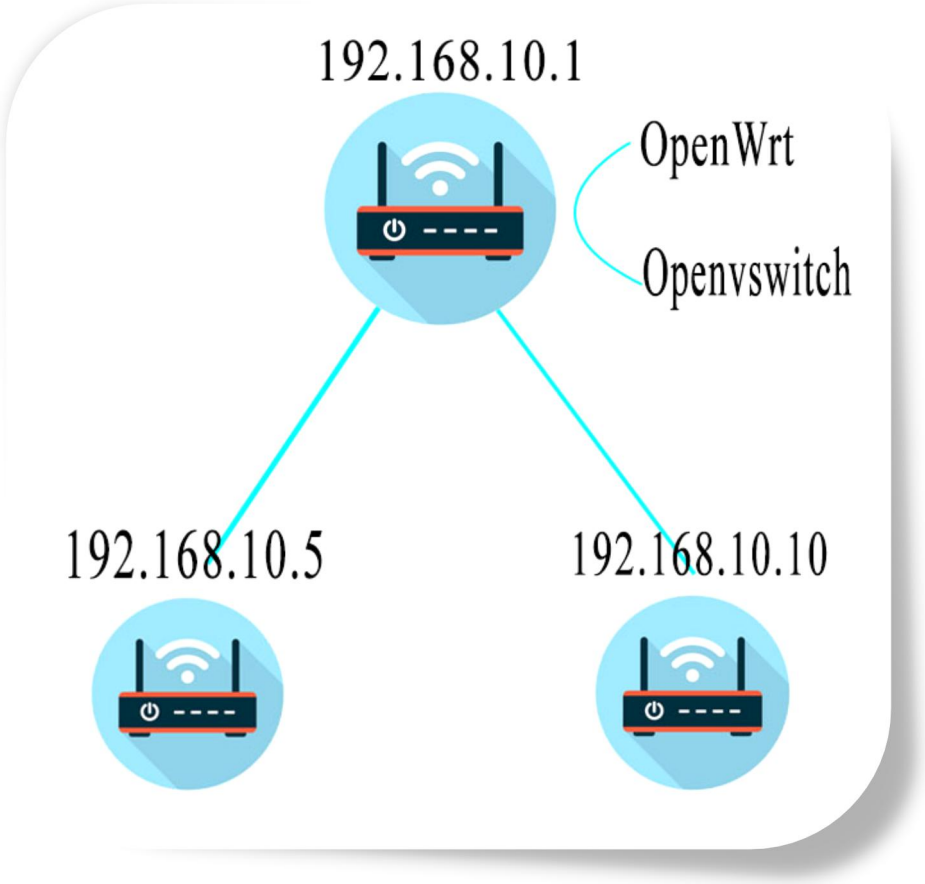
Bước 4: Cắm cáp RJ45 nối vào cổng LAN trên máy tính và cổng LAN trên router

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.

3.1.4 Lựa chọn hình trạng mạng

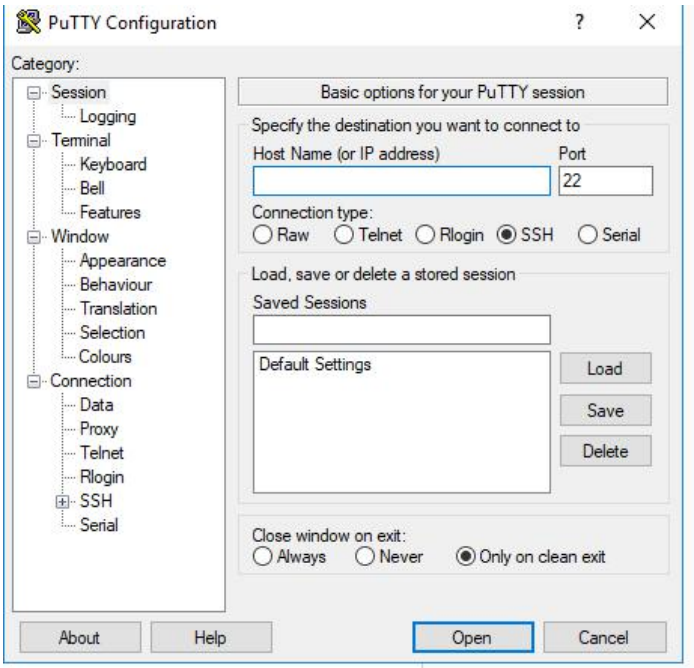
Chúng tôi cân nhắc lựa chọn một hình trạng mạng phổ biến đối với mạng Wi-Fi và quyết định chọn một router Wi-Fi có cấu hình tốt để làm gateway cho nhánh mạng và mở rộng bằng 2 router có cấu hình trung bình như hình 3.4



Hình 3.4. 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.



Hình 3.5. 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

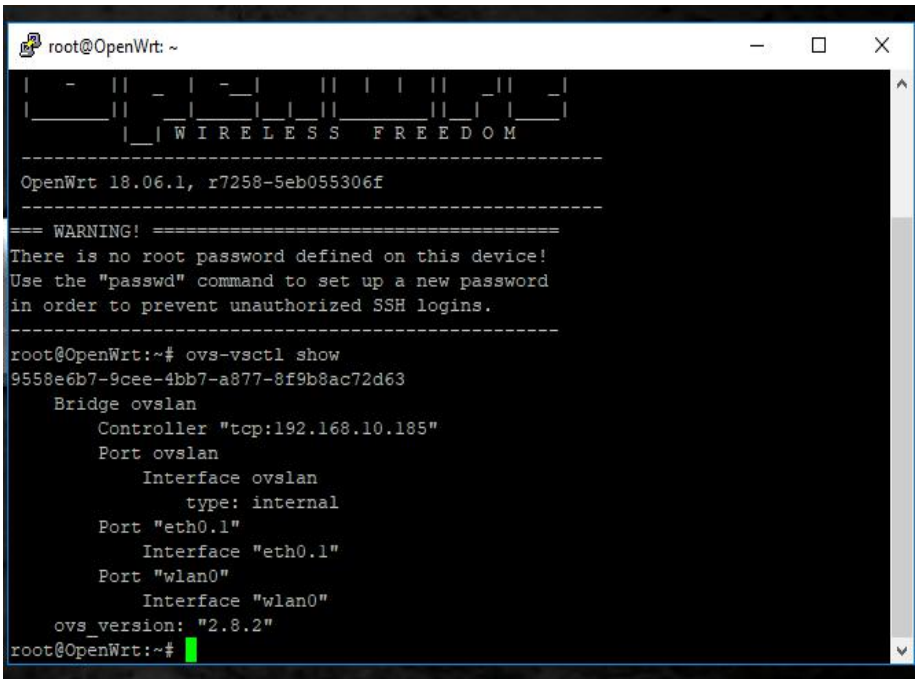
giám sát và điều khiển luồng dữ liệu thông qua cầu nối này. Để tạo cầu nối bằng OpenVSwitch ta sử dụng lệnh sau:

#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>



Hình 3.6. 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.

3.1.5 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.

3.1.6 Thiết kế giải thuật cân bằng tải

Như đã đề cập ở mục 3.14, 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

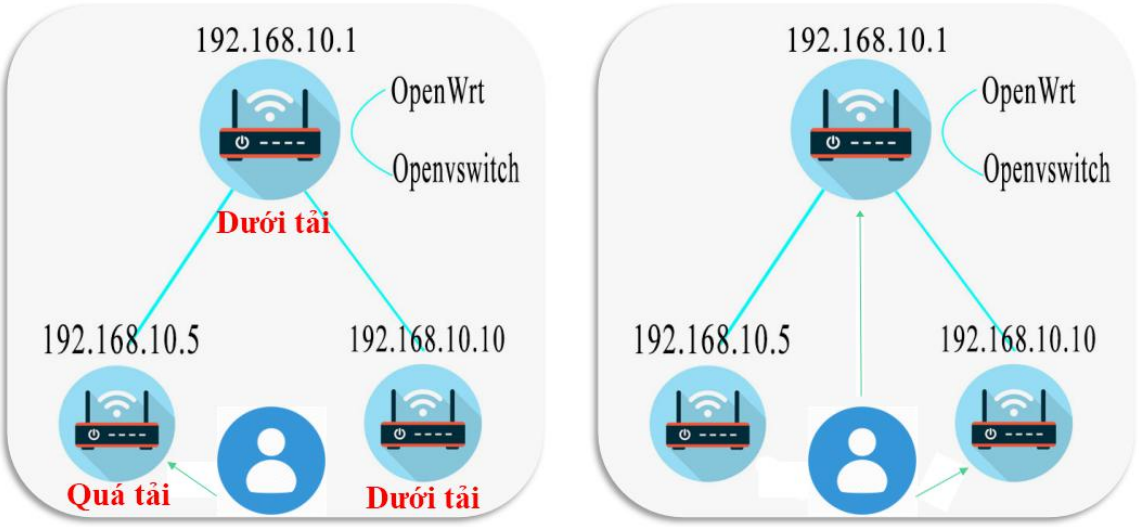
Tiếp theo, chúng tôi định nghĩa mức tải đối với router là mức sử dụng CPU ở thời điểm hiện tại và mức quá tải là mức CPU cao nhất mà router có thể hoạt động bình thường. Thêm nữa, chúng tôi cũng định nghĩa mức cảnh báo là mức CPU mà router có khả năng quá tải nếu có thêm nối kết đến nó.

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.



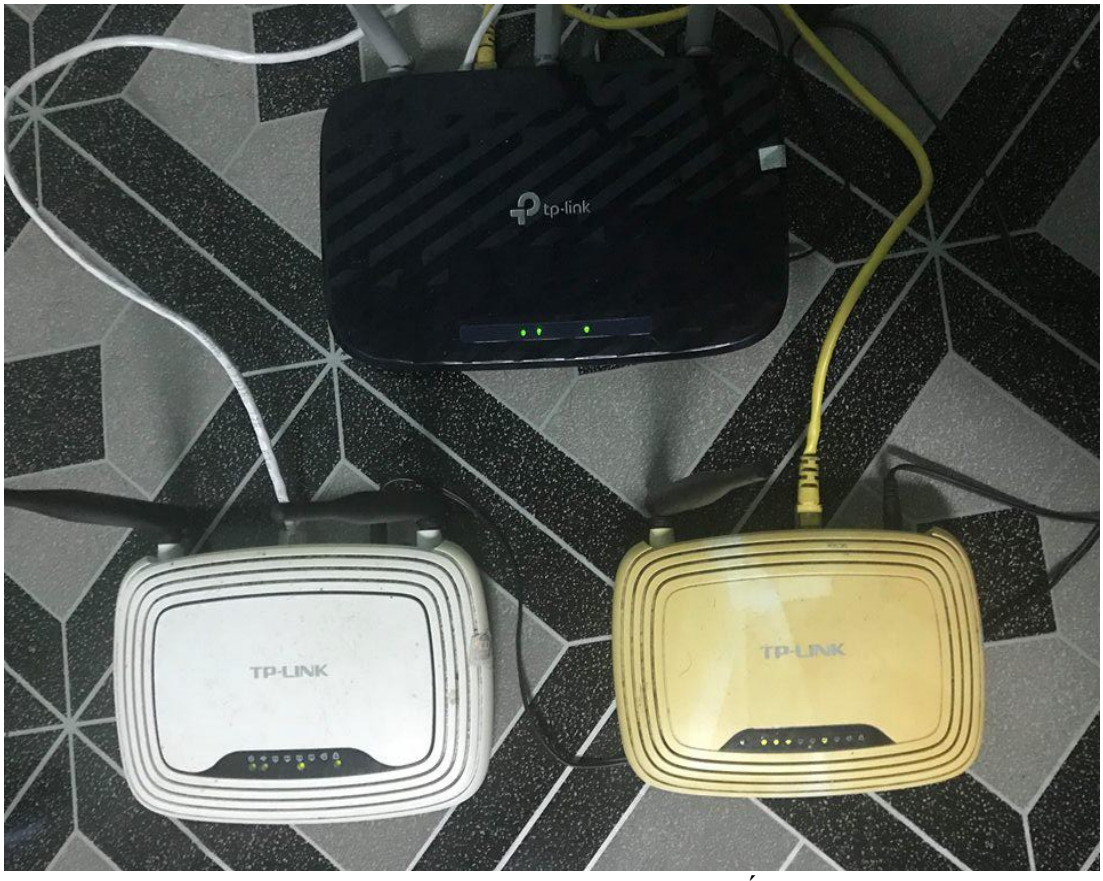
Hình 3.7. Ý tưởng chuyển client

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:

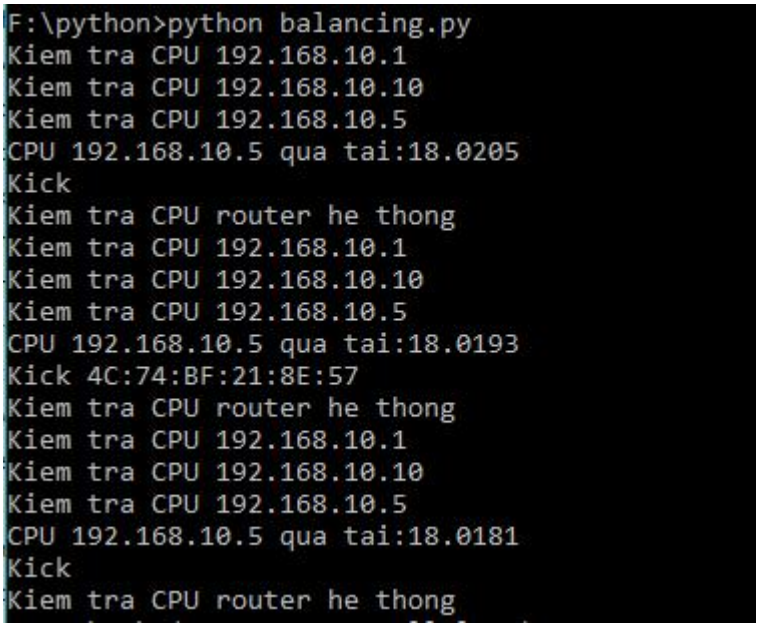
- Một router chính tp-link acher c20 có cài cả firmware OpenWrt và openvswitch có khả năng kết nối với controller để nhận các lệnh điều khiển dòng dữ liệu và điều khiển nối kết.
- Hai router con tplink-741 và tplink-841 được cài firmware OpenWrt có khả năng chia sẽ dữ liệu và nhận điều khiển từ controller.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



Hình 3.8. 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



Hình 3.9. Màn hình xử lý cân bằng tải

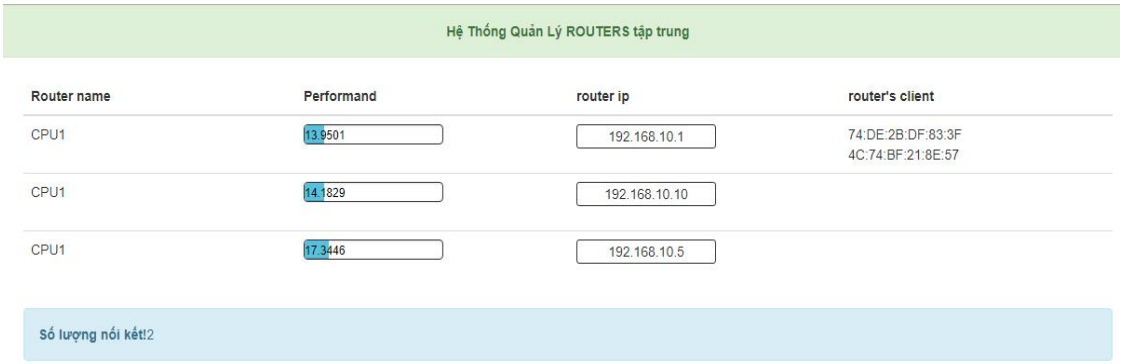
Phần controller điều khiển dữ liệu bằng giao thức openflow trên framework ryu cũng được áp dụng thành công.

```
C:\Users\WING\Desktop\ryu-master\ryu\app>ryu-manager switch.py
loading app switch.py
loading app ryu.controller.ofp_handler
instantiating app switch.py of SimpleSwitch13
instantiating app ryu.controller.ofp_handler of OFPHandler
packet in 189686267582337 74:de:2b:df:83:3f f8:d1:11:41:3b:5d 3
packet in 189686267582337 f8:d1:11:41:3b:5d 74:de:2b:df:83:3f 1
packet in 189686267582337 74:de:2b:df:83:3f ac:84:c6:99:17:81 3
packet in 189686267582337 74:de:2b:df:83:3f ac:84:c6:99:17:81 3
packet in 189686267582337 74:de:2b:df:83:3f f8:d1:11:41:3b:5d 3
packet in 189686267582337 74:de:2b:df:83:3f ac:84:c6:99:17:81 3
packet in 189686267582337 74:de:2b:df:83:3f ac:84:c6:99:17:81 3
packet in 189686267582337 ac:84:c6:99:17:81 74:de:2b:df:83:3f 4294967294
packet in 189686267582337 ac:84:c6:99:17:81 74:de:2b:df:83:3f 4294967294
packet in 189686267582337 ac:84:c6:99:17:81 74:de:2b:df:83:3f 4294967294
packet in 189686267582337 ac:84:c6:99:17:81 74:de:2b:df:83:3f 4294967294
packet in 189686267582337 4c:74:bf:21:8e:57 ac:84:c6:99:17:81 3
packet in 189686267582337 ac:84:c6:99:17:81 4c:74:bf:21:8e:57 4294967294
packet in 189686267582337 74:de:2b:df:83:3f ac:84:c6:99:17:81 3
```

Hình 3.10. Điều khiển luồng dữ liệu bằng Ryu

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.



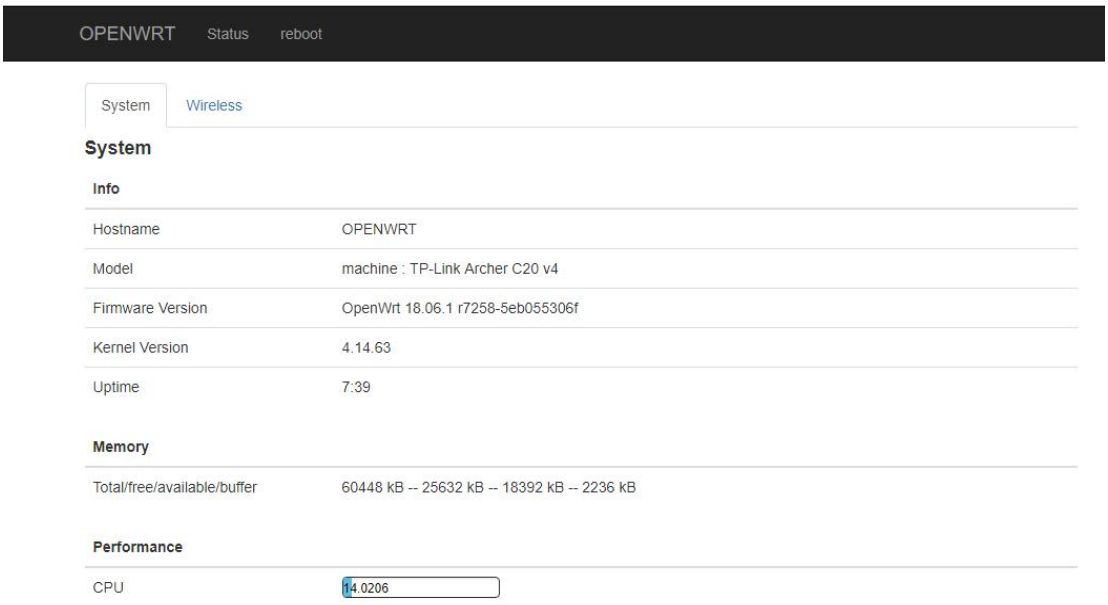
Hình 3.11. Màn hình quản lý chung trên website

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

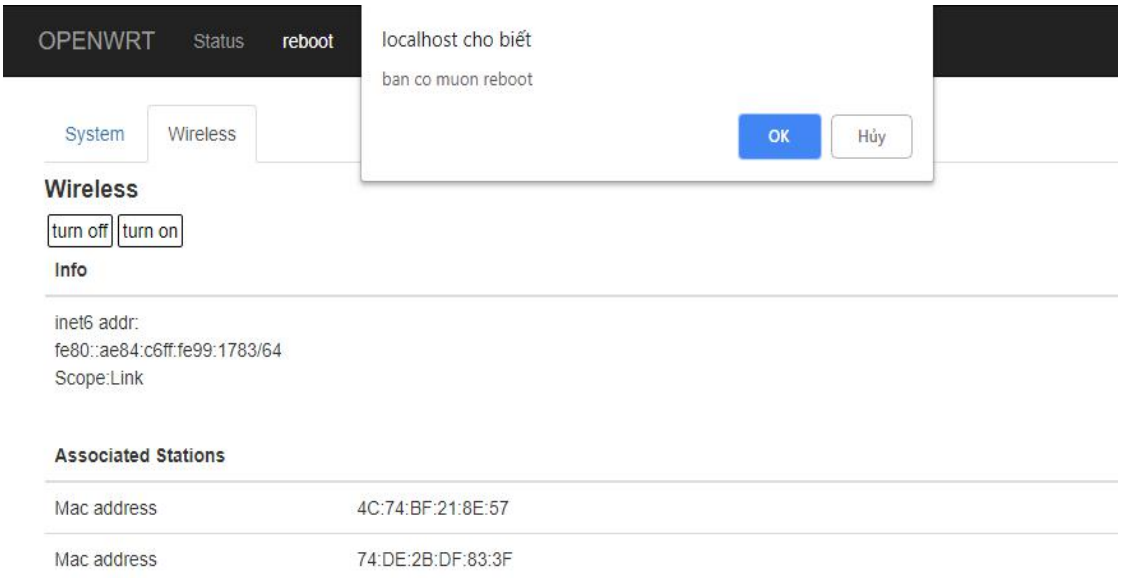
(Vì các router đang trong tình trạng ổn định nên thông số ngưỡng cpu được đặt ở mức 15% để kiểm tra code)

Hình 3.12. Màn hình 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.



Hình 3.13. Các thông tin cơ bản của router Wi-Fi



Hình 3.14. Thông tin wireless và các nối kết

CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 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 hoàn 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.

4.2 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.

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

[2] http://vientin.com

[3] https://vi.wikipedia.org/wiki/Wi-Fi

[4] https://trangcongnghe.com

[5] http://www.pcworld.com.vn

[6] https://vi.wikipedia.org/wiki/OpenWrt

[7] https://bachkhoa-aptech.edu.vn/

[8] http://www.academia.edu

[9] http://cdit.ptit.edu.vn

[10] https://github.com

[11] https://ryu.readthedocs.io

[12] https://viblo.asia

PHỤ LỤC