BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**Đề tài**

**CÂN BẰNG TẢI HỆ THỐNG WI-FI**

**BẰNG BỘ ĐIỀU KHIỂN TẬP TRUNG**

**Sinh viên thực hiện:**

**Lê Trung Tín-B1401101**

**Phạm Quốc Khải-B1401149**

**Giáo viên hướng dẫn:**

**Ts.Thái Minh Tuấn**

**Cần Thơ, 11/2018**

**LỜI CẢM ƠN**

MỤC LỤC

[CHƯƠNG 1: GIỚI THIỆU 5](#_Toc529455015)

[1.1 Đặt vấn đề 5](#_Toc529455016)

[1.2 Mục tiêu đề tài 5](#_Toc529455017)

[1.3 Đối tượng và phạm vi nghiên cứu 5](#_Toc529455018)

[1.4 Cơ sở nghiên cứu khoa học 5](#_Toc529455019)

[1.5 Phương pháp nghiên cứu 5](#_Toc529455020)

[1.6 Bố cục luận văn 7](#_Toc529455021)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 8](#_Toc529455022)

[2.1 Tóm tắt chương 8](#_Toc529455023)

[2.2 Các khái niệm, định nghĩa 8](#_Toc529455024)

[CHƯƠNG 3. PHƯƠNG PHÁP NGHIÊN CỨU 15](#_Toc529455025)

[3.1 Tóm tắt chương 15](#_Toc529455026)

[3.2 Giới thiệu quy trình nghiên cứu 15](#_Toc529455027)

[3.3 Phương pháp nghiên cứu 16](#_Toc529455028)

[CHƯƠNG 4. KẾT QUẢ VÀ ĐÁNH GIÁ 27](#_Toc529455029)

[CHƯƠNG 5.KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 28](#_Toc529455030)

# BẢNG TỪ VIẾT TẮT

# CHƯƠNG 1: GIỚI THIỆU

## Đặt vấn đề

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 nghiêm trọng cần giải quyết như:

* Nhiễu kênh truyền: Các router Wi-Fi do không “biết” được kênh hoạt động của nhau nên thường xuyên xảy ra tình trạng các router khác nhau hoạt động trên cùng một kênh truyền làm ảnh hưởng nghiêm trọng đến hiệu suất và thông lượng của Wi-Fi.
* 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 đề tài

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, điều khiển hiệu năng, danh sách các thiết bị nối kết, kênh 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.

## Đối tượng và phạm vi nghiên cứu

**Đối tượng tượng nghiên cứu:**

* Router Wi-Fi.
* Firmware OpenWrt.

**Phạm vi nghiên cứu:**

* Nội dung: luận văn tập trung nghiên cứu về hoạt động chia sẻ tài nguyên của các thiết bị sử dụng Wi-Fi
* Thời gian: 4 tháng kể từ tháng 8 năm 2018
* Không gian: đề tài được nghiên cứu trong một căn phòng diện tích khoảng 15 mét vuông có 3 router Wi-Fi.

## Cơ sở nghiên cứu khoa học

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

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.

Nghiên cứu cách cấu hình các nhánh mạng.

## Bố cục luận văn

* **Chương 1: Giới thiệu**
* **Chương 2: Cơ sở lý thuyết**
* **Chương 3: Phương pháp thực hiện**
* **Chương 4: Thực nghiệm và đánh giá**
* **Chương 5: Kết luận và hướng phát triển**

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

## Wi-Fi

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 hoặc 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 moderm 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.[3]

**Phương pháp roaming**

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.

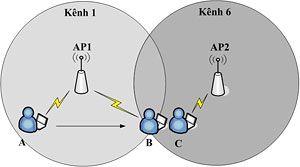
Cách 2: khi client di chuyển ra cận vùng phủ sóng router, tín hiệu yếu, không đảm bảo kết nối, nó sẽ tiến hành tìm kiếm router khác gần nhất.

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.

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. [4]

## OPENWRT**:**

Là một hệ điều hành nhúng dựa trên nhân Linux, và chủ yếu được sử dụng trên các thiết bị nhúng vào mạng lưới định tuyến đường truyền. Các thành phần chính là hạt nhân Linux, util-linux, uClibc và BusyBox. Tất cả các thành phần đã được tối ưu hóa kích thước, phải đủ nhỏ để nhúng vào bộ nhớ có hạn của các router dùng trong gia đình.

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.

OpenWrt có thể chạy trên các bộ định tuyến CPE, hệ thống mạng nhỏ, điện thoại thông minh và cả máy tính xách tay, máy tính bàn. OpenWrt có sự mở rộng rất linh hoạt và đa dạng tính năng.

* OpenWrt quản lý các gói cài đặt phần mềm của mình bằng phần mềm quản lý gói opkg. OPKG cho phép người dùng cài đặt và gỡ bỏ phần mềm. Các kho gói 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.
* OpenWrt có hầu hết các tính năng mạng như: Định tuyến thông qua iproute2, Quagga, BIRD,…
* Tích hợp các chức năng không dây hoàn hảo: có thể sử dụng router cài đặt OpenWrt như một repeater không dây, một điểm truy cập không dây, một cầu nối không dây, một cổng khóa, hoặc một sự kết hợp của những việc này với ví dụ ChilliSpot, Wi-FiDog Captive Portal, vv
* OpenWrt bao gồm các chuẩn bảo mật tin cậy: Packet injection, ví dụ: Airpwn, lorcon, ea, tường lửa, NAT và port forwarding thông qua netfilter; IP tunneling mở rộng mạng lưới giám sát thời gian thực và thống kê thông qua các ví dụ: RRDtool, Collectd, Nagios, Munin lite, Zabbix, …
* 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; thiết bị có thể được ví dụ như kết nối qua USB với một thiết bị nhúng bao gồm máy in băng thông rộng di động, modem webcam, card âm thanh

OpenWrt có các gói phần mềm đáng chú ý để sử dụng các phần cứng hỗ trợ là:

* Hệ thống chia sẻ tập tin qua SAMBA, (tương thích với Windows), NFS và FTP, chia sẻ máy in qua máy chủ CUPS in (spooling) hoặc p910nd (không spooling)
* PulseAudio, Music Player Daemon, Audio / Video trực tuyến thông qua các tiêu chuẩn AV DLNA / UPnP, iTunes (DAAP) máy chủ Asterisk (PBX), MQ Telemetry Transport qua Mosquitto
* Mở rộng một giao diện Web Ajax, nhờ dự án Luci, fix 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.

Môi trường và xây dựng phát triển hệ thống OpenWrt, được biết đến với nhau như OpenWrt Buildroot, được dựa trên một hệ thống Buildroot thay đổi rất nhiều. OpenWrt Buildroot là một tập hợp các bản vá lỗi và Makefile Đó Tự động hóa Quy trình xây dựng một hệ thống hoàn chỉnh dựa trên Linux OpenWrt cho một thiết bị nhúng, bởi Xây dựng và sử dụng một Cross-compilation toolchain thích hợp.

Thiết bị nhúng thường sử dụng một bộ xử lý khác nhau hơn so với một máy tính được tìm thấy trong máy chủ được sử dụng để xây dựng hình ảnh hệ thống OpenWrt của họ, đòi hỏi một toolchain cross-biên dịch. Như một toolchain lập chạy trên một hệ thống máy chủ, nhưng tạo ra mã cho một thiết bị nhúng nhắm mục tiêu và kiến trúc tập lệnh của vi xử lý (ISA). Ví dụ, nếu một hệ thống máy chủ sử dụng x86 và một hệ thống mục tiêu sử dụng MIPS32, toolchain compilation thường xuyên của các máy chủ chạy trên x86 và tạo ra mã cho kiến trúc x86, trong khi các toolchain cross-compilation chạy trên x86 và tạo ra mã cho các kiến trúc MIPS32. OpenWrt Buildroot Tự động hóa toàn bộ quá trình này để làm việc trên các kiến trúc tập lệnh của Embedded Devices Hầu hết các máy chủ và hệ thống

OpenWrt Buildroot Cung cấp các tính năng làm cho nó dễ dàng để port phần mềm trên các kiến trúc khác nhau. Sử dụng kconfig (Linux kernel menuconfig) cho cấu hình của tất cả các tùy chọn, cung cấp một toolchain cross-compiler tích hợp (gcc, ld, uClibc vv), cung cấp một khái niệm trừu tượng autotools (automake, autoconf), cmake và SCons xử lý tiêu chuẩn OpenWrt hình ảnh xây dựng quy trình làm việc: tải về, vá, cấu hình, biên dịch và đóng gói, cung cấp một số bản sửa lỗi phổ biến cho biết gói nặng hành xử.

### 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 là viết tắt của Remote Procedure Calls là giao thức gọi thủ tục từ xa được mã hóa bằng JSON. Nó là một giao thức rất đơn giản (và rất giống với XML-RPC), chỉ định nghĩa một vài kiểu dữ liệu và các lệnh. JSON-RPC cho phép thông báo ( dữ liệu gửi đến máy chủ và không yêu cầu phản hồi) và cho nhiều cuộc gọi được gửi đến máy chủ có thể được trả lời không đúng thứ tự.

Thủ tục gọi hàm từ xa JSON-RPC-API là phần cốt lỗi để xây dựng website điều khiển tập trung.

**JSON-RPC-API và cách sử dụng**

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[5],](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.

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 tệp cấu hình ban đầu (được chương trình đọc) theo cài đặt đã chọn trong tệp 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 tệp cấu hình ban đầu của chương trình bị ghi đè. Ngoài ra, tệp 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 tệp 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 tệp 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 tệp cấu hình bằng trình soạn thảo văn bản hoặc sửa đổi chúng bằng chương trình tiện ích dòng lệnh uci. Các tệp 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 tệp UCI.

Khi thay đổi tệp cấu hình UCI, cho dù thông qua trình soạn thảo văn bản hoặc dòng lệnh, dịch vụ hoặc tệp 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 tệp cấu hình phần mềm tiêu chuẩn cụ thể của họ. Tập lệnh init.d đầu tiên viết đúng một tệp cấu hình vào vị trí dự kiến ​​của phần mềm và nó được đọc lại bằng cách khởi động lại tệp thực thi. Lưu ý rằng chỉ (re) 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 tệp cấu hình dự kiến của chương trình. Những thay đổi trong các tập tin trong / etc / config / sau đó không có hiệu lực.

### HOSTAPD

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.[6] https://w1.fi/hostapd/

### 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.

### 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.

Topology

Phần cứng

Router acher c20

Cap xoắn đôi

# CHƯƠNG 3. PHƯƠNG PHÁP NGHIÊN CỨU

## Giới thiệu quy trình nghiên cứu

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

Giải thích quy trình nghiên cứu

Bước 1: Thu thập thông tin

Sau khi xác định vấn đề nghiên cứu, bước đầu tiên của quy trình là tìm hiểu thông tin của các phần cứng, phần mềm giải thuật liên quan đến đề tài. Khảo sát chọn lọc tài liệu thích hợp với đề tài cũng như xác định phương pháp thực hiện khả thi nhất.

Bước 2:sau khi nghiên cứu tài liệu tìm được các phần cứng phần mềm tiến hành cài đặt phần mềm, flash firmware cho router xây dựng topology, xây dựng giải thuật cùng cotroller và giao diện web.

Bước 3: đánh giá và kết luận

Tiến hành đánh giá hiệu năng của hệ thống, kiểm tra băng thông và xác định độ hiệu quả của giải thuật,

## Phương pháp thực hiện

3.3.1 nghiên cứu tài liệu chọn giải pháp thích hợp

Để thực hiên đề tài “cân bằng tải Wi-Fi dựa trên hệ thống điều khiển tập trung” cần xây dựng một topology thích hợp cho hệ thống, cần kết nối các router lại với nhau theo các chuẩn mô hình mạng như là(blabla).

Đối với một hệ thống mạng thông thường có thể chọn bất cứ lại router nào trên thị trường đề xây dựng một topology thích hợp. Tuy nhiên hệ thống đòi hỏi cần phải cài đặt các router có thể giao tiếp trao đổi thông tin với nhau nên không thể sử dụng các router và phần mềm mặc định của nhà cung cấp.

Giải pháp đưa ra là cài đặt mới một “hệ điều hành” khác cho router đề có thể thực hiện các truy vấn và cho phép người quản trị có thể can thiệp sâu cũng như có toàn quyền điều khiển router. Hệ điều hành có thể thực hiện đầy đủ các yêu cầu trên là OpenWrt.

3.3.2 Chọn router,firmware,flash firmware

**Chọn router**

Lý do chọn router

**Topology**

Vẽ hình

Lý do

**Cách thức thực hiện**

3.3.3 Xây dựng website điều khiển tập trung

3.3.3.1 Lý do xây dựng website.

3.3.3.2 các bước thực hiện

**Ý tưởng để xây dựng website theo dõi hệ thống router**

Để xây dựng website theo dõi tập trung nhóm sẽ sử dụng ngôn ngữ lập trình php gửi các yêu cầu resful để nhận được các thông tin phản hồi mong muốn.

API được cung cấp bởi OpenWrt không có đầy đủ mọi thứ cần thiết mà website yêu cầu vì vậy nhóm sẽ sử dụng phương pháp lập trình bash shell, cùng các lệnh (command line) của OpenWrt để xây dựng một số script cần thiết cho website này(script và các lệnh này có thể sử dụng kiến trúc restful để gọi ra và nhận về các thông tin cần thiết bằng php).

Các dữ liệu được trả về sau khi gửi thông điệp restful là các chuỗi 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 thông số ram, cpu

Thì không thể sử dụng theo cách gửi restful thông thường.Ý tưởng đưa ra là sử dụng ajax của javascript để gửi lại các yêu cầu restful theo một chu kỳ nhất định.

Ngoài ra một trong những chức năng chính của website là cấu hình cho 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 hơn cho lập trình viên.

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

**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 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 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 chương phần này sẽ sử dụng tập lệnh uci để cấu hình cho các router. Có thể truy cập vào router qua ssh và thực hiện kiểm tra cách thức hoạt động của các lệnh uci.

VD: Để bật chức năng phát Wi-Fi từ tập lệnh uci ta có thể thực hiện các lệnh sau:

uci set wireless.@Wi-Fi-iface[0].ssid="tênWi-Fi"

uci set wireless.@Wi-Fi-iface[0].encryption=psk2

uci set wireless.@Wi-Fi-iface[0].key="mậtkhẩuWi-Fi"

uci commit

service network reload

Như vậy sau khi biết cách sử dụng thư viện uci ta có thể xây dựng chức năng cấu hình cho Wi-Fi như sau

Xây dựng tập lệnh cấu hình Wi-Fi bằng uci .

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 các command đã xây dựng

Chờ cho router thực hiện các lệnh và khởi động lại hệ thống 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.

Để website có thể quản lý toàn bộ thông tin của các router nhóm đã tạo ra một cơ sở dữ liệu trên server lưu lại toàn bộ các thông tin cần thiết như ip, tên.

Dựa trên cơ sở dữ liệu này mỗi router sẽ được cấp cho một token riêng. Từ các token này những hàm truy vấn thông tin bằng restful sẽ được gọi ra và trả về các thông tin của toàn bộ router trong hệ thống mạng.

Các thông tin của router sẽ được sắp xếp cho người quản trị có thể dễ dàng quan sát và quản lý hệ thống mạng bằng website tập trung

Thêm vào đó website quản lý tập trung sẽ có thêm các chức năng thêm, sửa, xóa các router nếu hệ thống mạng ngoài đời thực được thay đổi.

Sau khi xây dựng website có sơ đồ cây chức năng như sau:

Vẽ hình

3.3.4 Xây dựng controller

Nhiệm vụ của controller là:

Kiểm tra tình trạng tải của toàn bộ

router trong hệ thống đảm bảo nó luôn trong tình trạng dưới tải.

Giới hạn lại số lượng thiết bị nối kết khi cần thiết

Kiểm soát toàn bộ các nối kết của mạng router phân bổ lại nối kết theo giải thuật cài đặt.

**Chức năng cân bằng tải.**

**Chức năng gán kênh tự động.**

Sơ đồ dòng dữ liệu giải thuật:

# CHƯƠNG 4. KẾT QUẢ VÀ ĐÁNH GIÁ

## 4.1 kết quả

Sau quá trình thực hiện đề tài nhóm đã thu được kết quả như sau:

Xây dựng được mô hình topology 3 router được cài đặt firmware OpenWrt. Mô hình mạng này có thể chia sẽ thông tin về tình trạng hoạt động, phiên làm việc, lưu lượng dữ liệu vào ra của chúng.

Xây dựng được website kiểm soát tập trung cho phép theo dõi tình trạng của hệ thống mạng, tính trạng hoạt động của router, hiện những cảnh báo thích hợp khi các router xuất hiện tình trạng quá tải. Ngoài ra người quản trị còn có thể sử dụng website để cấu hình các router trong hệ thống qua các chức năng cấu hình của website.

Xây dựng thành công controller-bộ điều khiển tập trung được xây dựng bởi các giải thuật kiểm soát mạng có thể đưa ra các quyết định chuyển nối kết, giới hạn người dùng hệ thống mạng và một số quyết định khác để đảm bảo tình trạng tải của từng router cũng như toàn bộ hệ thống mạng.

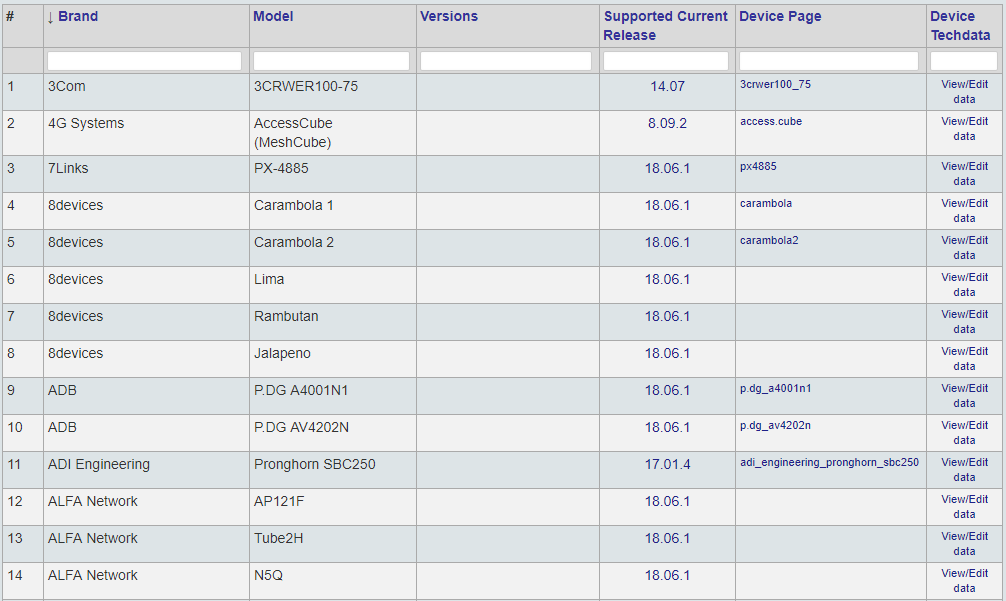
## 4.2 đánh giá

# CHƯƠNG 5.KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

# PHỤ LỤC

Firmware OpenWrt là một hệ điều hành hoàn hảo cho các router tuy nhiên nó chỉ cung cấp cho một số router nhất định trên thị trường chứ không phải tất cả.

Có thể tham khảo các thiết bị mà firmware này hỗ trợ ở trang web chính thức của nó: OpenWrt.org

Để dễ dàng tiện dụng hơn cho việc cài đặt thêm các gói phần mềm, mở rộng chức năng cho router sau khi cài đặt hệ điều hành nên chọn các router có bộ nhớ flash lớn hơn 8mb

**Chọn version firmware**

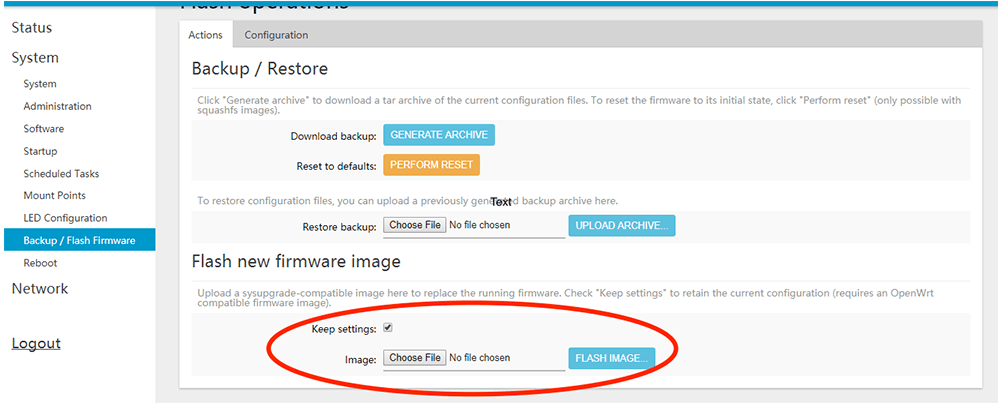
Sau khi chọn router cần chọn firmware thích hợp. Các firmware được chia ra nhiều version thích hợp với từng loại router mà nó hỗ trợ.





**Cài đặt hệ điều hành**

Ở một số router ta có thể cài đặt firmware OpenWrt một cách trực tiếp thông qua giao diện web mặc định cung cấp bởi nhà sản xuất

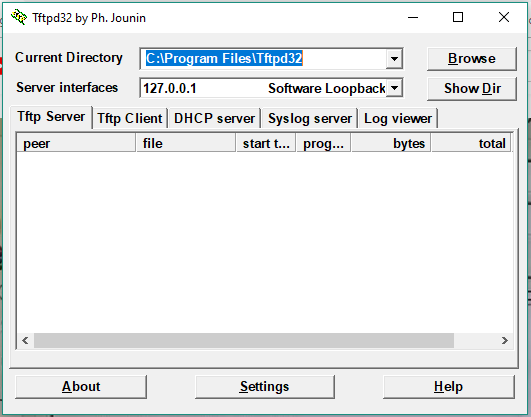
Tuy nhiên ở một số router khác nhà sản xuất sẽ chặn việc cập nhật firmware OpenWrt bằng giao diện. Bởi vậy ta phải thực hiện việc cài đặt firmware này bằng cách flash firmware qua công nghệ tftp.

Ý tưởng của công nghệ tftp:Khi router được bật nguồn, bootloader sẽ chuẩn bị khởi động cho các chương trình cơ sở đồng thời khởi tạo các cổng lan có dây, trong vài giây router sẽ lắng nghe các tftp request(yêu cầu cho tftp) nếu nhận được một ảnh(image) từ tftp server để cài đặt nó sẽ bắt đầu quá trình cài đặt firmware cho router, nếu không nhận được gì nó sẽ bắt đầu nạp chương trình trong ROM và khởi động bình thường.

Các bước thực hiện flash firmware:

Chuẩn bị:

Phần mềm tftpd



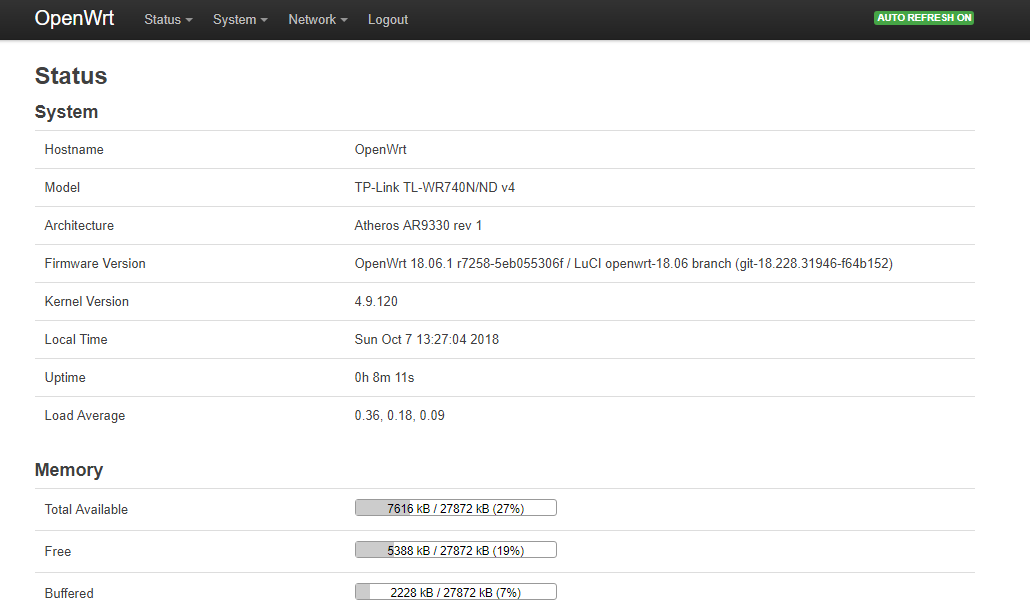
File cài đặt(firmware OpenWrt)



Thực hiện:

* Khởi động chương trình tftpd
* Nhấn browse tìm đến thư mục chứa file cài đặt firmware
* Cắm cap router vào máy tính
* Khởi động router
* Chờ đến khi router nhận được file cài đặt từ tftpserver
* Đợi cho thanh process cài đặt chạy đến hết

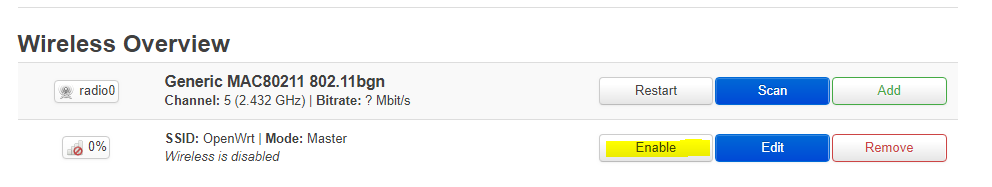
Sau khi cài đặt xong truy cập địa chỉ 192.168.1.1 để vào giao diện ban đầu của router.



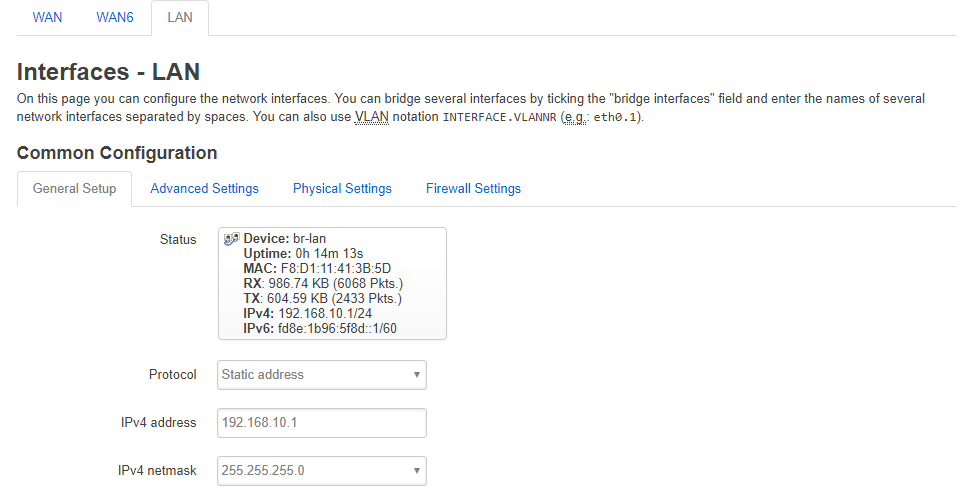
**Cấu hình sơ bộ cho router**

Để xây dựng một topology hoàn chỉnh cần cấu hình router một cách phù hợp để tránh xung đột ip và tạo ra một nhánh mạng như ý muốn.

Bật wireless



Đổi ip



Cài đặt lua-mod-rpc: OpenWrt cho phép người dùng có thể sử dụng dịch vụ ssh để truy cập vào Wi-Fi như một hệ thống linux. Người dùng có thể cài đặt các gói bằng giao diện dòng lệnh với trình quản lý gói opkg (có thể cài đặt các gói này bằng giao diện web).

