**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**



**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

─────── \* ───────



**BÁO CÁO**

Đồ án môn học chuyên ngành TTM

**Đề tài:** Tìm hiểu về SDN, kiến trúc của ONOS, xây dựng mạng sử dựng kiến trúc ONOS

*Sinh viên thực hiện:*

|  |  |  |
| --- | --- | --- |
| **Họ Và Tên** | **MSSV** | **Lớp** |
| Nguyễn Công Hưng | 20131945 | CNTT1.02-K58 |

*Giáo Viên hướng dẫn*: **TS. Trần Hải Anh**

**Hà Nội, 5-2017**

**Mục Lục**

[Lời Nói Đầu 3](#_Toc483467365)

[Chương 1.Tổng quan về SDN 4](#_Toc483467366)

[1. SDN là gì? 4](#_Toc483467367)

[2. Kiến trúc của SDN 4](#_Toc483467368)

[3. Lợi ích của SDN 6](#_Toc483467369)

[3.1 Tập trung hóa việc điều khiển trong môi trường mạng của nhiều nhà cung cấp 6](#_Toc483467370)

[3.2 Giảm độ phức tạp thông qua tự động hóa 6](#_Toc483467371)

[Chương 2. Tìm hiểu về ONOS 8](#_Toc483467372)

[1. Tổng quan về ONOS 8](#_Toc483467373)

[2. Kiến trúc của ONOS 8](#_Toc483467374)

[2.1 Dịch vụ và hệ thống con (Services and Subsystems) 9](#_Toc483467375)

[2.3 Các nhà cung cấp (Provider) 11](#_Toc483467376)

[2.4 ProviderID 11](#_Toc483467377)

[2.5 Multiple Providers 12](#_Toc483467378)

[2.6 Manager 12](#_Toc483467379)

[2.7 Store 12](#_Toc483467380)

[2.8 Application 13](#_Toc483467381)

[2.9 ApplicationID 13](#_Toc483467382)

[2.10 Events and Descriptions 13](#_Toc483467383)

[2.11 Network representations 15](#_Toc483467384)

[Chương 3: Cài đặt môi trường ONOS 16](#_Toc483467385)

[Chương 4: Xây dựng mạng sử dụng ONOS 19](#_Toc483467386)

[1. Yêu cầu đặt ra 19](#_Toc483467387)

[2. Thực hiện yêu cầu 19](#_Toc483467388)

[2.1 Chuẩn bị thiết bị cần thiết 19](#_Toc483467389)

[2.2 Xây dựng mạng 19](#_Toc483467390)

# Lời Nói Đầu

Cuộc cách mạng khoa học hiện đại lần thứ 4 đang được hình thành và phát triền. Trong cuộc cách mạng này không thể thiếu được sự tham gia chặt chẽ của hệ thống các máy tính và đặc biệt là hệ thống mạng máy tính thế hệ mới cần được phát triển và hoàn thiện để tham gia vào cuộc cách mạng khoa học hiện đại lần này.

SDN là một kiến trúc mạng thế hệ mới góp phần tham gia mạnh mẽ cho cuộc cách mạng khoa học lần thứ tư. SDN được xây dựng với những đặc điểm mới dễ thích nghi với điều kiện hệ thống mạng ngày nay, ngoài ra nó có nhưng tính năng phù hợp như: năng động, dễ dàng trong quản lý, dễ thích nghi, phù hợp với hệ thống mạng có băng thông cao hiện nay, và thích hợp với tính năng động của các ứng dụng mới hiện nay.

Với những đặc điểm đó của SDN, nên trong môn học Đồ án chuyên ngành mạng TTM, em quyết định tìm hiểu về SDN và kiến trúc của ONOS để học hỏi về SDN, sau đó em triển khai một mạng với kiến trúc của ONOS.  
 Với dự học hỏi và cố gắng của bản thân cũng như sự chỉ bảo tận tình của TS Trần Hải Anh, em đã hoàn thành được đồ án môn học TTM. Do thời gian làm có hạn cũng như trình độ của em còn hạn chế nên không tránh khỏi những thiếu sót. Em rất mong nhận được sự đóng góp ý kiến của các thầy cô đề hoàn thiện bài báo cáo này. Cuối cùng em xin cảm ơn tới thầy Trần Hải Anh (giảng viên Viện Công nghệ thông tin và truyền thông) đã chỉ bảo và giúp đỡ em hoàn thành được bài báo cáo này. Em xin chân thành cảm ơn.

# Chương 1.Tổng quan về SDN

## SDN là gì?

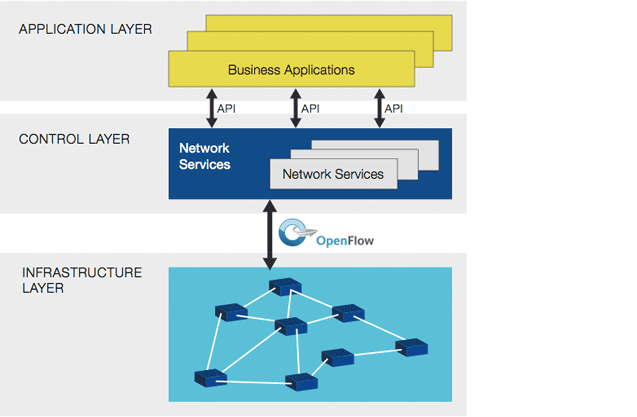
Hầu hết các mạng thông thường đều theo kiến trúc phân cấp, được xây dựng với các tầng của thiết bị chuyển mạch Ethernet, được sắp xếp theo cấu trúc cây. Thiết kế này thực sự hiệu quả khi mô hình tính toán khách–chủ chiếm ưu thế, nhưng kiến ​​trúc cố định như vậy không thích hợp với yêu cầu tính toán đa dạng, năng động và nhu cầu lưu trữ dữ liệu tại các trung tâm dữ liệu của doanh nghiệp, trường học và trong môi trường của các nhà cung cấp dịch vụ. Một số xu hướng tính toán quan trọng dẫn tới yêu cầu ngày càng tăng cho một mô hình mạng mới bao gồm: Sự thay đổi mô hình lưu lượng; Hướng tới người dùng CNTT; Sự phát triển của các dịch vụ điện toán đám mây; “Dữ liệu lớn” yêu cầu nhiều băng thông hơn.

SDN là tên viết tắt của Software-Defined Networking có nghĩa là mạng điều khiển bằng phần mềm được dựa trên cơ chế khai phá tách bạch việc kiểm soát một luồng mạng với luồng dữ liệu. SDN dựa trên giao thức luồng mở (Open Flow) và là kết quả nghiên cứu của Đại học Stanford và California, Berkeley. SDN tách định tuyến và chuyển các luồng dữ liệu riêng rẽ và chuyển kiểm soát luồng sang thành phần mạng riêng có tên gọi là thiết bị kiểm soát luồng (Flow [Controller](http://vlo.vn/tag/controller/)). Điều này cho phép luồng các gói dữ liệu đi qua mạng được kiểm soát theo lập trình.

SDN cũng bao gồm khả năng ảo hóa các nguồn lực mạng. Nguồn lực mạng ảo hóa được biết đến như một “ngăn mạng” (network slice). Một ngăn có thể mở rộng nhiều thành phần mạng bao gồm đường trục mạng, bộ định tuyến và các host. Khả năng kiểm soát nhiều luồng dữ liệu sẽ tạo ra sự linh hoạt và nguồn lớn hơn trong tay người sử dụng.

## Kiến trúc của SDN

Kiến trúc SDN bao gồm: Lớp ứng dụng (Application Layer), lớp điều khiển (Control Layer) và lớp hạ tầng cơ sở (Infrastructure Layer).



***Lớp ứng dụng*** cung cấp các giao diện có khả năng lập trình mở; Phần mềm Virtual Cloud Networks cho phép các nhà cung cấp dịch vụ điện toán đám mây cung cấp các dịch vụ đám mây công cộng tự động và có khả năng mở cho các doanh nghiệp. Sử dụng phần mềm này, các tổ chức, doanh nghiệp có thể tạo ra một “đám mây ảo” cô lập, thông qua hạ tầng cơ sở đám mây công cộng tự phục vụ, giúp họ sự kiểm soát hoàn toàn cho các dịch vụ và các ứng dụng mới cho người sử dụng.  
Cũng tại lớp ứng dụng, phần mềm ứng dụng mới Sentinel Security tự động kiểm soát truy cập và bảo đảm phòng chống xâm nhập cho các mạng trong khuôn khổ của tổ chức, doanh nghiệp với phần cứng chuyển đổi OpenFlow thông qua bộ điều khiển. Có thể giảm bớt sự phức tạp và chi phí của các thiết bị phần cứng chuyên dụng, đồng thời có được sự đảm bảo khả năng mở rộng cần thiết cho các ứng dụng mới.

***Lớp điều khiển*** cung cấp cách nhìn tập trung và sự tự động cấu hình mạng của tấtcả các thiết bị trong hạ tầng cơ sở. Bộ điều khiển cho phép các nhà quản trị mạng thiết lập chương trình một cách dễ dàng, linh hoạt và mở rộng môi trường mạng của họ cho các ứng dụng tự động cảm ứng đơn. Nó cũng cung cấp các giao diện chương trình ứng dụng (APls) cho các nhà phát triển bên thứ ba để tùy chỉnh tích hợp các ứng dụng.

***Lớp hạ tầng cơ sở*** cung cấp việc truy cập có khả năng lập trình mở thông qua OpenFlow, một giao thức mạng giúp tự động cấu hình phần cứng. Chức năng SDN mới trong lớp hạ tầng cơ sở cho phép người quản trị đơn giản cấu hình mạng, mang đến một giao diện linh hoạt và khả năng lập trình theo tiêu chuẩn.

## Lợi ích của SDN

Mạng SDN dựa trên OpenFlow cho phép giải quyết các vấn đề liên quan tới băng thông, sự thay đổi liên tục của các ứng dụng, chuyển đổi mạng cho phù hợp với các yêu cầu làm việc và giảm đáng kể độ phức tạp của hoạt động điều hành và quản lý mạng. Các lợi ích có thể đạt được thông qua kiến ​​trúc mạng SDN dựa trên OpenFlow bao gồm:

* 1. Tập trung hóa việc điều khiển trong môi trường mạng của nhiều nhà cung cấp

Phần mềm điều khiển SDN có thể kiểm soát thiết bị mạng hỗ trợ OpenFlow từ bất kỳ nhà cung cấp nào, bao gồm chuyển mạch, định tuyến và chuyển mạch ảo. Thay vì phải quản lý từng nhóm thiết bị từ các nhà cung cấp riêng lẻ, nhà quản lý có thể sử dụng đồng bộ các thiết bị và các công cụ quản lý dựa trên SDN để nhanh chóng triển khai, cấu hình và cập nhật các thiết bị trong toàn bộ mạng.

### **Giảm độ phức tạp thông qua tự động hóa**

Mạng SDN dựa trên giao thức OpenFlow cung cấp các công cụ giúp tự động hóa và quản lý mạng một cách linh hoạt. Điều này hỗ trợ nhà quản lý có thể phát triển các công cụ giúp thực hiện các tác vụ quản lý một cách tự động hóa.  
 3.3 Tốc độ đổi mới cao hơn

Sử dụng mạng SDN làm tăng khả năng đổi mới trong công việc, bằng cách cho phép nhà vận hành mạng có thể thực sự lập trình theo thời gian thực để đáp ứng những yêu cầu công việc đặc biệt và nhu cầu phát sinh của người sử dụng, bằng cách ảo hóa và trừu tượng hóa cơ sở hạ tầng mạng từ các dịch vụ mạng.  
 3.4 Tăng cường độ tin cậy và an ninh mạng  
 SDN cho phép các nhà quản lý tự định nghĩa các cấu hình cấp cao (high-level configuration) và chính sách trong mạng, điều này được chuyển xuống cơ sở hạ tầng thông qua OpenFlow. Kiến trúc mạng SDN dựa trên OpenFlow loại bỏ nhu cầu phải cấu hình lại cho từng thiết bị mạng đơn mỗi khi một thiết bị đầu cuối có sự thay đổi. Điều này làm giảm thiểu khả năng phát sinh lỗi trong mạng do xung đột cấu hình hoặc chính sách.   
Bộ điều khiển mạng SDN cung cấp khả năng hiển thị đầy đủ và kiểm soát qua mạng. Điều này đảm bảo rằng việc kiểm soát truy cập, lưu lượng, chất lượng dịch vụ, an ninh và các chính sách khác được thực thi nhất quán trên các cơ sở hạ tầng mạng của các tổ chức. Bởi vậy, sẽ giảm chi phí hoạt động, khả năng cấu hình linh hoạt, ít gặp lỗi, thực thi chính sách và cấu hình thống nhất.  
 3.5 Trải nghiệm người dùng tốt hơn

Bằng cách tập trung hóa điều khiển mạng và đảm bảo thông tin trạng thái sẵn sàng cho các ứng dụng cấp cao hơn, cơ sở hạ tầng mạng SDN có thể thích ứng tốt hơn với nhu cầu đa dạng của người dùng. Với mạng SDN dựa trên OpenFlow, các ứng dụng video có thể tự nhận diện băng thông cho phép trong mạng theo thời gian thực và tự động điều chỉnh độ phân giải video cho phù hợp.

# Chương 2. Tìm hiểu về ONOS

## Tổng quan về ONOS

ONOS (Open Network Operating System) là một hệ điều hành mạng dựa trên nền tảng SDN, nó cung cấp dịch vụ có khả năng mở rộng, tính sẵn sàng cao, hiệu năng cao và trừu tượng để tạo ra các ứng dụng và dịch vụ dễ dàng. Nền tảng này được dựa trên một kiến trúc vững chắc và đã nhanh chóng trưởng thành là tính năng phong phú và sẵn sàng được sử dụng.

ONOS được thiết kế để đáp ứng nhu cầu của các nhà cung cấp dịch vụ. Nó là hệ điều hành dựa trên các cụm (cluster), nó có tính mở rộng quy mô theo chiều ngang của mạng và nhu cầu của ứng dụng.Nó được chứng minh là có thể đạt được hiệu suất tối đa với lưu lượng dữ liệu truyền tải cao với độ trễ thấp.

Các tính năng mà ONOS cung cấp rất phù hợp cho lập trình trên thiết bị mạng:

-Tính sẵn sàng cao, khả năng mở rộng và hiệu năng rất lớn: Đây là một yêu cầu rất phù hợp với các nhà cung cấp và doanh nghiệp mạng trên thế giới

-Tính trừu tượng hóa và đơn giản mạnh: Yêu cầu này phù hợp cho việc phát triển ứng dụng và tìm kiếm các giải pháp phù hợp

-Giao thức và thiết bị độc lập với nhau: Tránh được những lỗi phát sinh do đặc điểm của giao thức

-Tách biệt các thành phần modun và hệ thống với nhau: Cho phép tùy chỉnh và sửa chữa không quan tâm đến hệ thống code như thế nào.

## Kiến trúc của ONOS

ONOS là một dự án multi-module, đó là những module được quản lý giống như một hệ thống OSGi (Open Services Gateway initiative). ONOS được xây dựng với nhưng mục tiêu chính là:

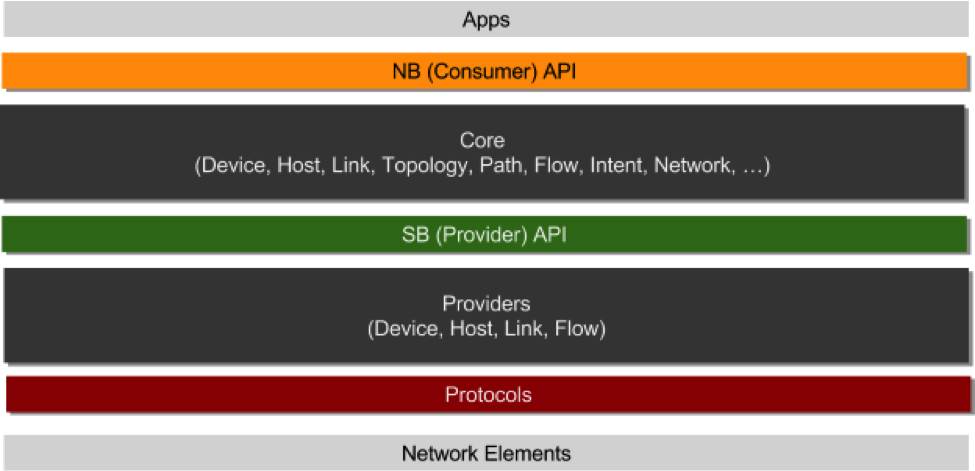
- Code Modularity (Modul mã): Nó có thể tạo ra các chức năng mới như là các thành phần khép kín.

- Configurability (Cấu hình): Nó có khả năng chạy và giải phóng các tính năng khác nhau kể cả khi khởi động hay khi chạy.

- Separation of Concern (Tách ranh giới rõ ràng): Tạo ranh giới rõ ràng cho hệ thống con(subsystems) để tạo điều kiện cho các module

- Protocol agnosticism (Giao thức không rõ ràng): Nó và ứng dụng của nó không bị ràng buộc với các thư viện giao thức cụ thể hoặc triển khai.

ONOS được xây dựng với các tằng mà ở đó mỗi tầng có những chức năng riêng. Hình ảnh bên dưới cho ta cái nhìn rõ ràng về kiến trúc ONOS.



### Dịch vụ và hệ thống con (Services and Subsystems)

Một dịch vụ (Services) là một đơn vị chức năng bao gồm nhiều thành phần tạo ra một phần theo chiều dọc thông qua các tầng như là một nhóm phần mềm. ONOS đã định nghĩa ra một số dịch vụ cơ bản:

-*Device Subsystem*: Quản lý lưu trữ hạ tầng của các thiết bị.

-*Link Subsystem*: Quản lý lưu trữ liên kết của thiết bị.

-*Host Subsystem*: Quản lý lưu trữ của các thiết bị đầu cuối và vị trí của nó trong mạng.

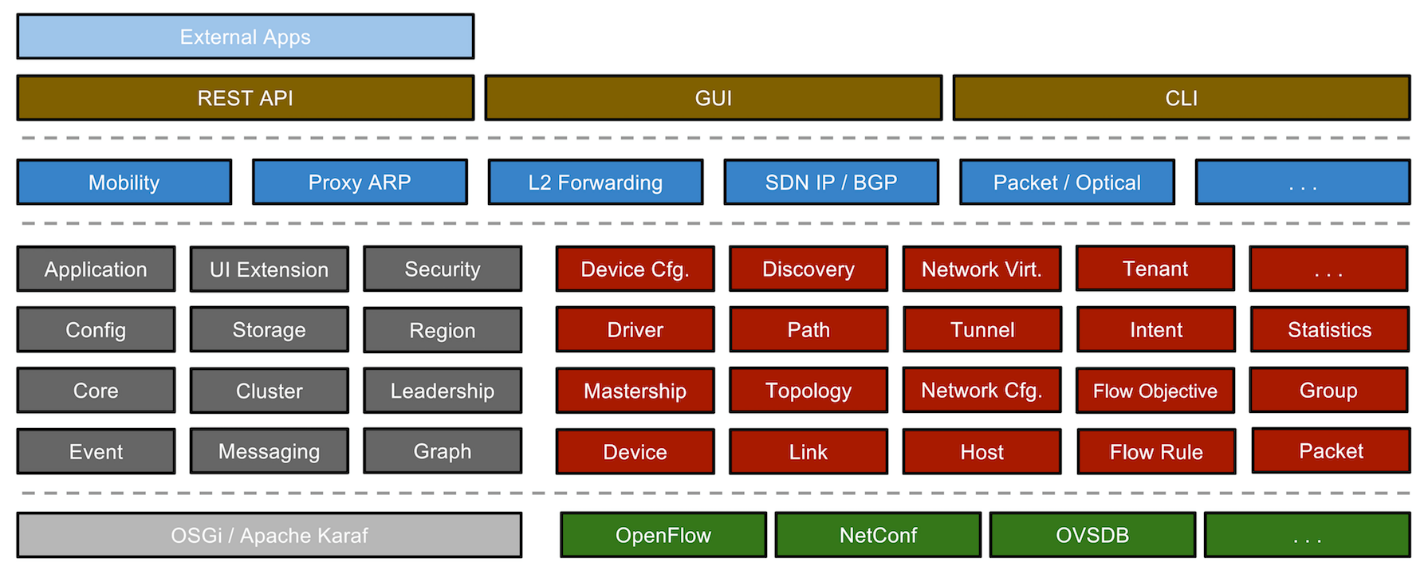
- *Topology Subsystem*: Quản lý mô hình mạng theo thời gian thông qua các biểu đồ.

-*PathService*: Các máy tính có thể tìm đường giữa hạ tầng thiết bị hoặc giữa các host sử dụng mô hình mạng gần nhất.

- *FlowRule Subsystem:* Quản lý kiểm kê các quy tắc luồng hay hành động đã được cài đặt trên thiết bị cơ sở hạ tầng và cung cấp chỉ số lưu lượng.

- *Packet Subsystem:* Cho phép các ứng dụng lắng nghe các gói dữ liệu nhận được từ các thiết bị mạng và tạo ra các gói dữ liệu ra mạng thông qua một hoặc nhiều thiết bị mạng.

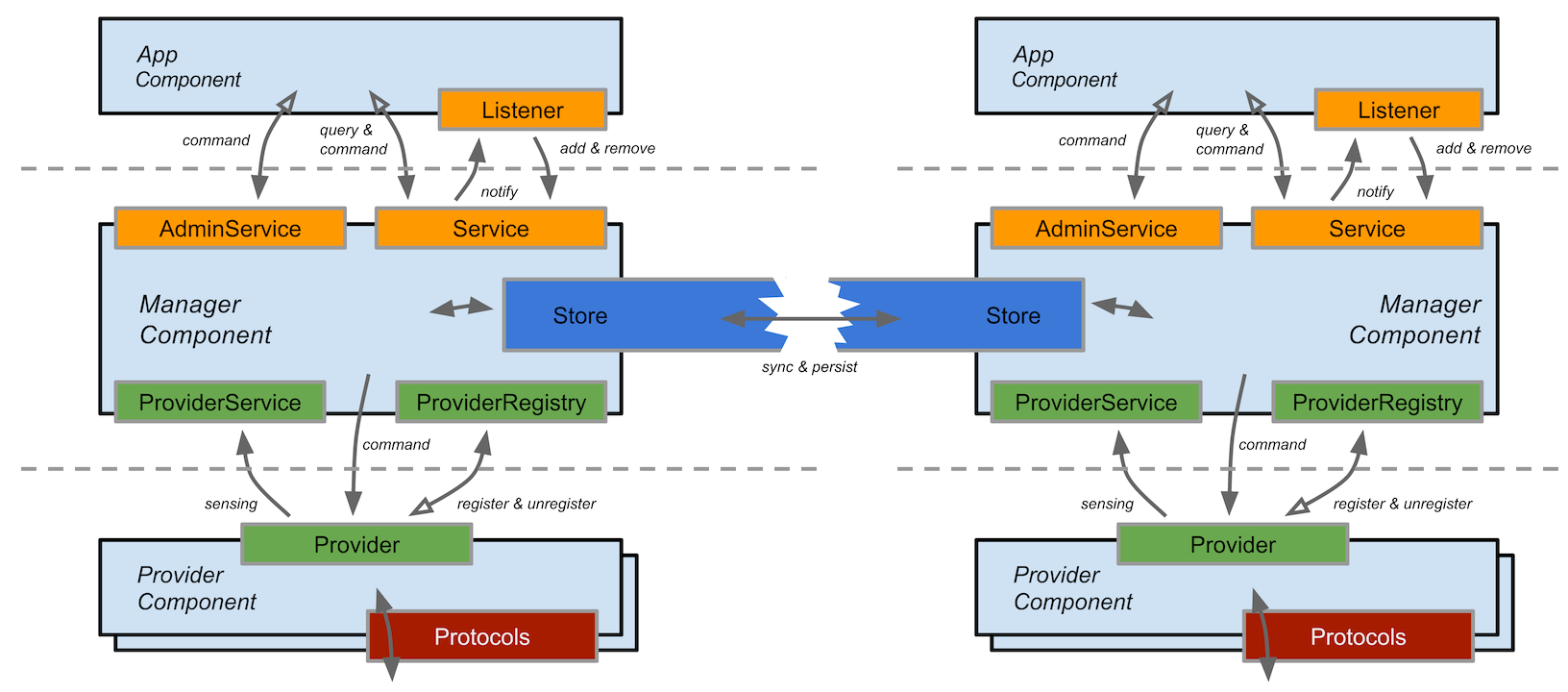
Hình bên dưới mô phỏng subsystem sẽ được xây dựng trong tương lại gần.



* 1. Cấu trúc của subsystem

Mỗi thành phần của một subsystem nằm trong một trong ba lớp chính và có thể được xác định bởi một hoặc nhiều interfaces Java mà chúng implement.

Hình dưới đây tóm tắt các mối quan hệ của các thành phần subsystem.



### Các nhà cung cấp (Provider)

Ở tầng thấp nhất của ngăn xếp ONOS, Các nhà cung cấp giao tiếp với mạng thông qua các thư viện cụ thể theo giao thức và với lõi thông qua *ProviderService* Interfaces.

Các nhà cung cấp dịch vụ nhận thức giao thức có trách nhiệm tương tác với môi trường mạng bằng cách sử dụng các giao thức điều khiển và cấu hình khác nhau và cung cấp các dữ liệu cảm giác dịch vụ cụ thể cho cốt lõi. Các nhà cung cấp cũng có thể thu thập dữ liệu từ các subsystem khác, chuyển chúng thành dữ liệu dịch vụ cụ thể.

Một số nhà cung cấp cũng có thể cần phải chấp nhận các sắc lệnh kiểm soát từ cốt lõi và áp dụng chúng vào mạng bằng cách sử dụng các phương tiện cụ thể cho từng giao thức. Chúng được đưa vào nhà cung cấp thông qua giao diện nhà cung cấp.

### ProviderID

Một nhà cung cấp được cấp 1 id riêng biệt. Mục đích chính của ProviderId là cung cấp một danh tính của một gia đình của các nhà cung cấp, cho phép các thiết bị và các thực thể mô hình khác vẫn liên kết với danh tính của nhà cung cấp chịu trách nhiệm về sự tồn tại của họ ngay cả sau khi nhà cung cấp được gỡ cài đặt hoặc dỡ bỏ.

ProviderId mang một thiết kế URI để cho phép ghép nối lỏng các thiết bị với các nhà cung cấp từ một nhà cung cấp gia đình thay thế và cho điều này có thể không có quyền truy cập vào Nhà cung cấp chính nó.

### Multiple Providers

Một subsystem thì được tổ chức thành nhiều các nhà cung cấp (Multiple Providers). Trong trường hợp này Provider được chỉ định giống như là chính hoặc phụ trợ. Provider chính sở hữu các thực thể liên kết với dịch vụ của mình, với các provider phụ trợ đóng góp thông tin của họ như lớp bao phủ. Cách tiếp cận này có ưu tiên hơn thông tin provider chính, nếu bất kỳ lớp bao phủ nào gây ra xung đột với thông tin lớp dưới.

Một device subsystem một dịch vụ được lưu trữ và hỗ trợ từ nhiều provider.

### Manager

Một thành phần nằm trong lõi, Manager nhận được thông tin từ các provider và phục vụ nó cho các ứng dụng và các dịch vụ khác. Nó có nhiều interface:

* Service interface hướng bắc thông qua đó các ứng dụng hoặc các thành phần cốt lõi khác có thể tìm hiểu về một khía cạnh cụ thể của trạng thái mạng
* AdminService interface để lấy các lệnh quản trị và áp dụng chúng vào trạng thái mạng hoặc hệ thống
* ProviderRegistry interface hướng nam thông qua đó Provider có thể đăng ký với Manager để có thể tương tác với nó
* ProviderService interface hướng nam được trình bày với Povider đã đăng ký, qua đó nó có thể gửi và nhận thông tin đến và từ Manager.

Một consumers của một Manager’service interface cso thể được nhận thông thin từ cả việc đồng bộ và không đồng củ các truy vấn, giống như các sự kiện lắng nghe (listener).

### Store

Nằm trong phần cốt lõi với Manager, Store có nhiệm vụ lập indexing, kiên trì và đồng bộ các thông tin nhận được từ Manager. Điều này bao gồm việc đảm bảo sự nhất quán và mạnh mẽ của thông tin trên nhiều trường hợp ONOS bằng cách trực tiếp giao tiếp với các store trên các trường hợp ONOS khác.

### Application

Các ứng dụng tiêu dùng và thao tác thông tin được tổng hợp bởi các Provider thông qua AdminService interface và Service Interface. Các ứng dụng có nhiều chức năng khác nhau, từ việc hiển thị topo mạng trong một trình duyệt web để thiết lập đường dẫn cho lưu lượng mạng.

### ApplicationID

Mỗi ứng dựng có duy nhất một ApplicationID. Định danh này được sử dụng bởi ONOS để theo dõi ngữ cảnh liên kết với một ứng dụng.

### Events and Descriptions

Hai đơn vị cơ bản của việc phân phối thông tin trong ONOS là sự kiện và Mô tả. Cũng giống như các dịch vụ, sự kiện và mô tả được kết hợp với các yếu tố và khái niệm mạng cụ thể. Cả hai đều bất biến khi được tạo ra.

#### Descriptions

Descriptions được sử dụng để truyền thông tin về một phần tử southbound API. Ví dụ: Một *HostDescription* bao gồm thông tin của mọt host và địa chỉ IP và cả địa chỉ của mạng. Descrptions thường được tạo thành từ một hoặc nhiều đối tượng mô hình, các mô tả của ONOS về các thành phần mạng khác nhau.

* + 1. Event

Sự kiện được sử dụng bởi Manager, để thông báo cho người nghe về những thay đổi trong mạng và bởi Store để thông báo cho các đồng nghiệp của họ về các sự kiện trong một cài đặt phân phối. Ví dụ *DeviceEvent* có thể được sử dụng để tạo thông báo *DeviceListener* là tạo thêm thiết bị (Device\_ADD) hay hủy bỏ một thiết bị (Device\_Remove) hay những trường hợp khác…

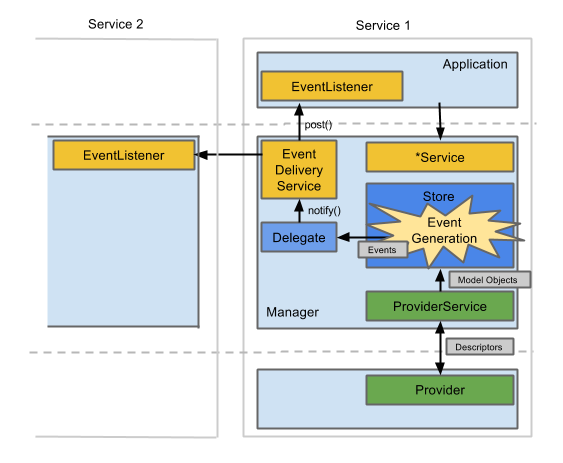
**Event dispatch**

Sự kiện được tạo ra bởi Store, dựa trên đầu vào Manager. Sau khi tạo, một sự kiện được gửi đến listener qua StoreDelegate interface, cuối cùng gọi đến EventDeliveryService. Bản chất của *StoreDelegate* chuyển các sự kiện ra khỏi Store, và *EventDeliveryService* đảm bảo sự kiện sẽ chỉ hướng đến Listener. Bởi chúng tương tác với nhau nên hai thành phần này được nằm tại Manager, nơi mà manager cung cấp lớp thực hiện là *StoreDelegate* tới store.

**Event Listeners**

Event Listeners được thực hiện ở *EventListener* interface. *EventListener* interface con phân lọa theo sự kiện mà nó lắng nghe. Phương thức điển hình của việc triển khai là dành cho một listener event là lớp bên trong của provider hoặc ứng dụng, từ đó các dịch vụ thích hợp được gọi dựa trên sự kiện đã nhận. Điều này hạn chế việc xử lý các sự kiện bên ngoài một subsystem tới subsytem manager hoặc ứng dụng, tức là các vị trí hợp lý nơi chúng cần được xử lý.

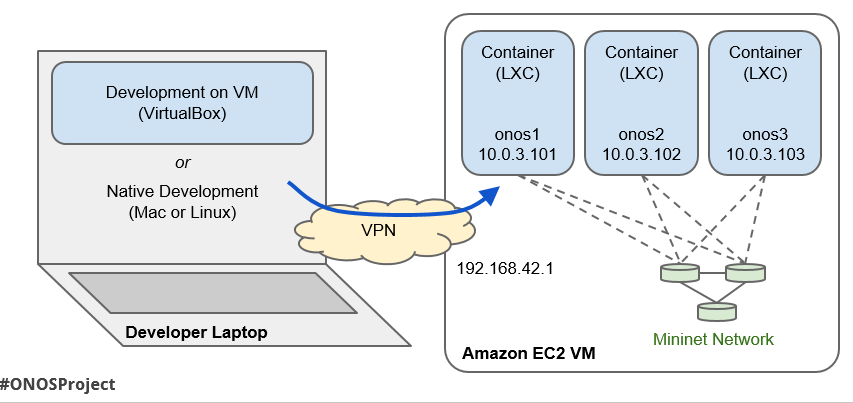
Hình ảnh bên dưới mô ta rõ nhất sự tương tác giữa event vs descriptions, Application và Provider.

****

### Network representations

Các mô hình đối tượng là các biểu diễn giao thức Agnostic của ONOS về các thành phần mạng và thuộc tính khác nhau. Các sự kiện mang những biểu tượng này như là cơ thể chủ thể của chúng. Các biểu diễn này được xây dựng từ các thông tin tìm thấy trong mô tả bởi các lõi của hệ thống.

# Chương 3: Cài đặt môi trường ONOS



Để cài đặt và lập trình trên ONOS chúng ta cần cài đặt những thành phần sau, được cài đặt trên ubuntu 14.04 :

1. Install *IntelliJ* (or *Eclipse*)
2. Install *Oracle JDK 8*
3. Install *apache-maven*
4. Install *apache-karaf*
5. Install *curl*
6. *git clone https://gerrit.onosproject.org/onos*
7. Set up the ONOS *bash\_profile*
8. Build onos
9. *git clone* [*https://github.com/bocon13/onos-byon*](https://github.com/bocon13/onos-byon)

Chúng ta có thể tải các cài đặt các thành phần trên theo các đường link sau:

IntelliJ IDEA:

<https://www.jetbrains.com/idea/download/>

Eclipse:

<https://eclipse.org/downloads/>

Oracle JDK8 <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Apache Maven (available via brew): <https://maven.apache.org/download.cgi?Preferred=ftp://mirror.reverse.net/pub/apache/>

Apache Karaf (3.0.5):

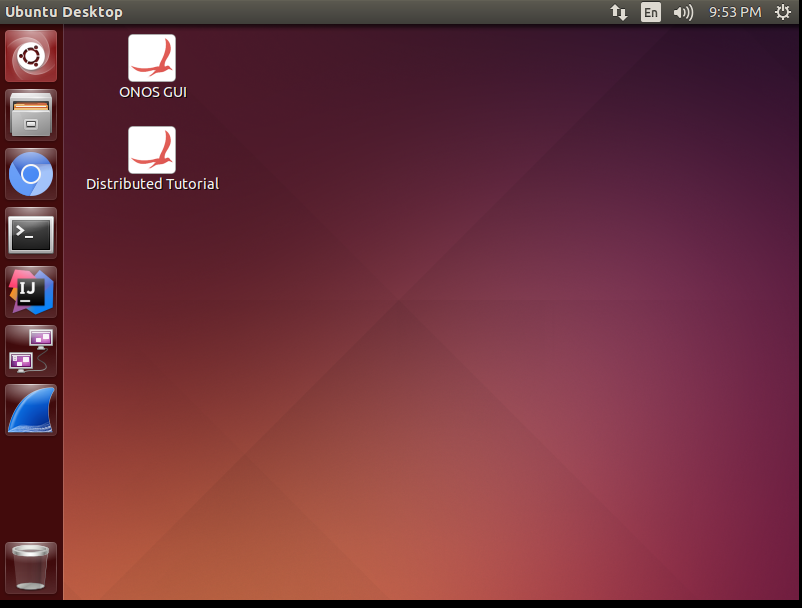
<https://karaf.apache.org/index/community/download.html>

cURL (available via brew):

<http://curl.haxx.se/download.html>

Sau khi cài đặt tất cả các thành phân trên chúng ta có thể bắt đầu làm việc với ONOS. Nếu không muốn mất công cài đặt các thành phần trung gian chúng ta có thể sử dụng máy ảo có sẵn do các nhà phát triển của onos phát hành.Chúng ta có thể tải về và add vào máy ảo VM Ware hay Virtual Box, Link tải ở đây: <https://downloads.onosproject.org/vm/ONOS-Tutorial-Falcon.ova> account: onos, password: onos

Hình ảnh hệ điều hành sau khi cài đặt thành công:



# Chương 4: Xây dựng mạng sử dụng ONOS

## Yêu cầu đặt ra

*Yêu cầu*: Trong phần này chúng ta sẽ tạo ra một mạng được xây dựng trên ONOS. Trong phần này chúng ta sẽ tạo ra 1 mạng bao gồm các host và switch được kết nối với nhau và cùng được kết nối về 1 controler ONOS, sau đó kiểm tả các thông tin của hệ thống mạng.

## Thực hiện yêu cầu

### Chuẩn bị thiết bị cần thiết

Để tạo ra một mạng các thiết bị host và switch, sử dụng phần mềm mininet được cài đtặ trên một máy ảo ubuntu khác, máy ảo này sẽ được kết nối với máy ảo đã cài đặt ONOS.

Các thiết bị và phần mềm cần chuẩn bị:

-Phần mềm VM Ware 12

-2 Máy ảo ubuntu được cài trên VM Ware 12, 1 máy đã cài đặt onos ,1 máy cài đặt mininet, 2 máy được nối với nhau trong cùng một mạng (trường hợp này em sử dụng mạng 10.0.0.0)

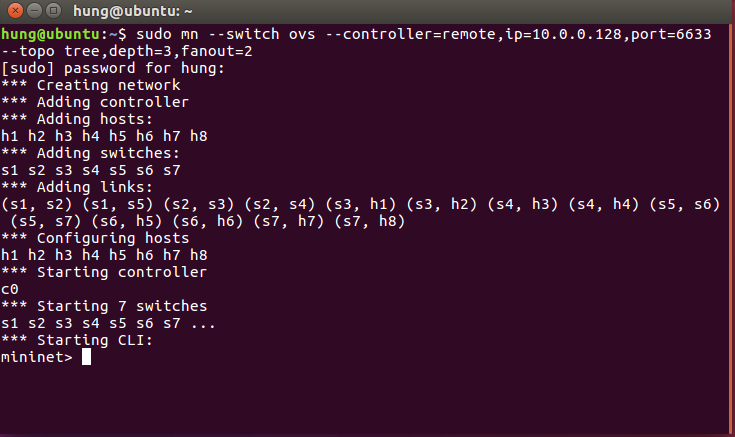
### Xây dựng mạng

Khởi động onos với câu lệnh: *onos-karaf install clear* .Đợi một khoảng thời gian xuất hiện giao diện như thế này:

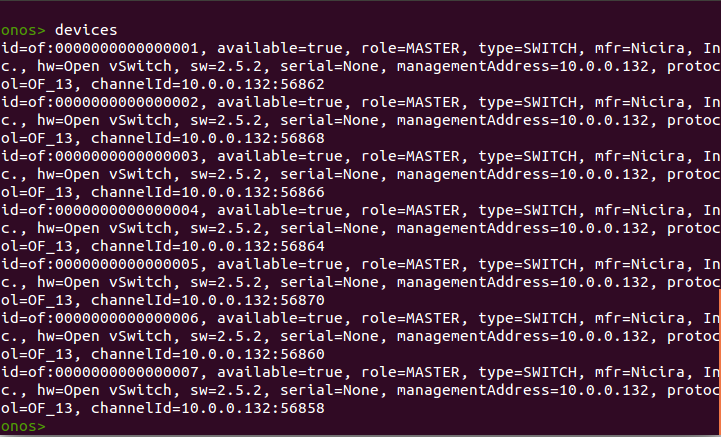


Trên máy ảo còn lại thực hiện câu lệnh tạo topo mạng mininet như sau: *sudo mn –switch ovs –controller remote, ip =10.0.0.128 --topo tree,depth=3, fanout=2*

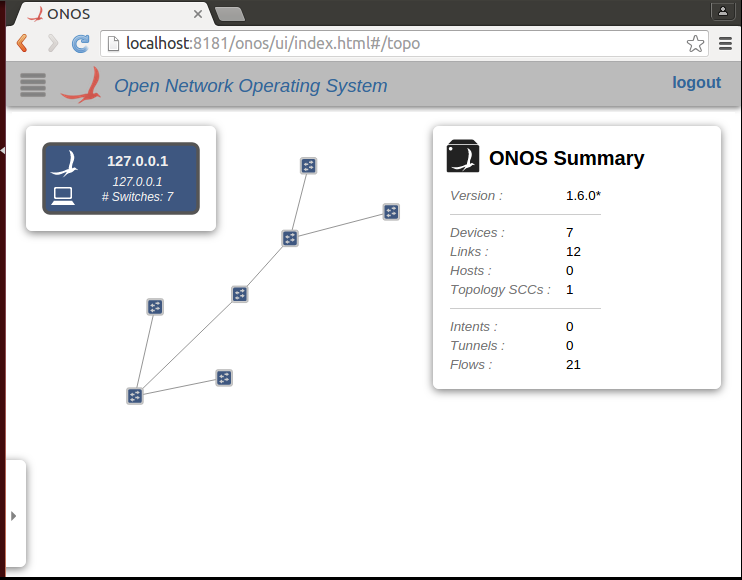
Trong lênh trên 10.0.0.128 là địa chỉ mạng của máy đã khởi động onos, 6633 là cổng mặc định của onos để connect vào, dạng topo mạng là tree với độ sâu là 3 và có số nút lá là 2 trên 1 switch, sau khi chạy lệnh này ta có hình ảnh như sau:



Để kiểm tra các thiết bị ta có thể gõ vào máy cài onos, từ devices để kiểm tra số lượng thiết bị kết nối trực tiếp vào onos.Hình ảnh bên dưới:

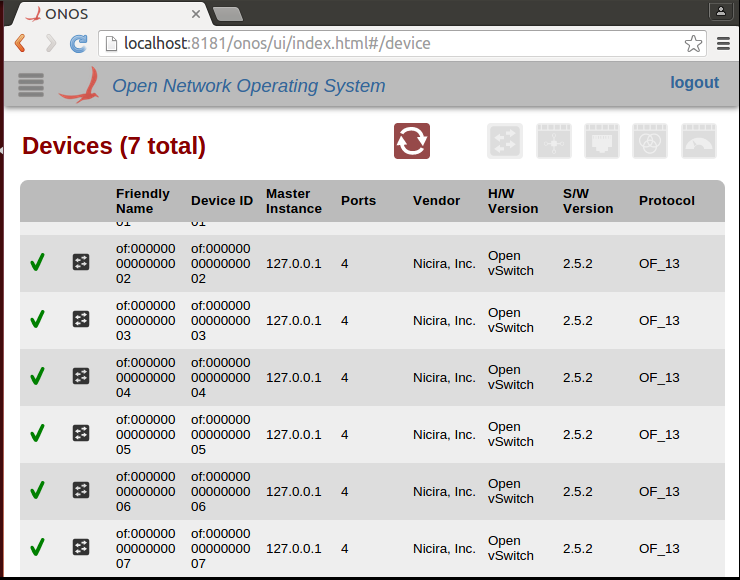


Để rõ hơn chúng ta vào máy cài onos truy nhập trình duyệt và gõ đoạn sau: <http://localhost:8181/onos/ui/login.html> . Đăng nhập với account: onos và password: rocks. Sau khi đăng nhập sẽ xuất hiện mô phỏng các switch kết nối trực tiếp với onos controller, hình ảnh như sau:

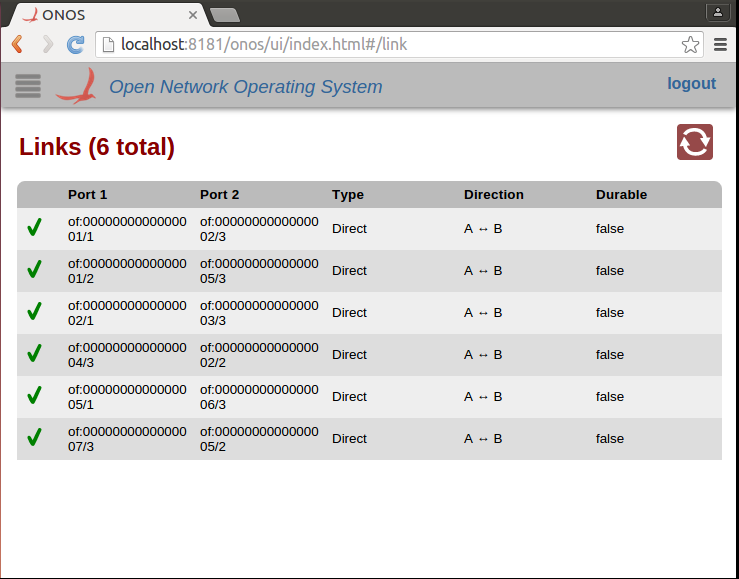


Kiểm tra các thông số dựa vào kí hiệu setting ở phía trên góc bên trái, chúng ta có thể kiểm tra các thông số của topo

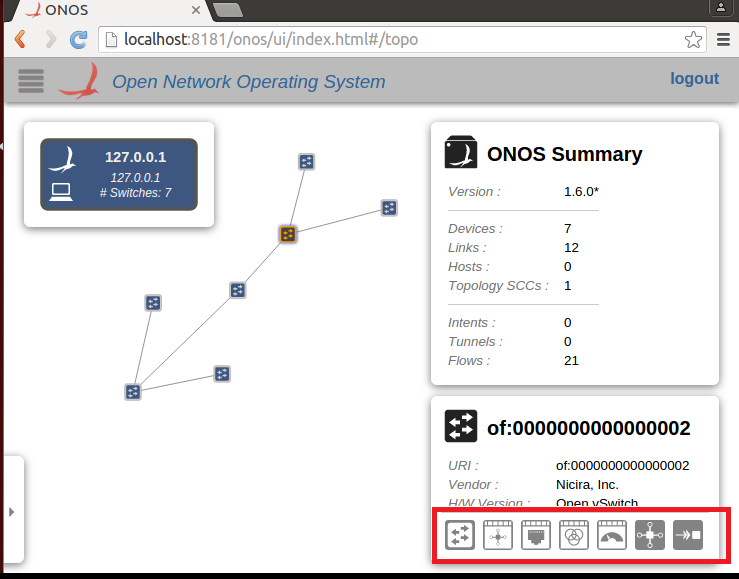
Kiểm tra các thiết bị:



Kiểm tra các kết nối:

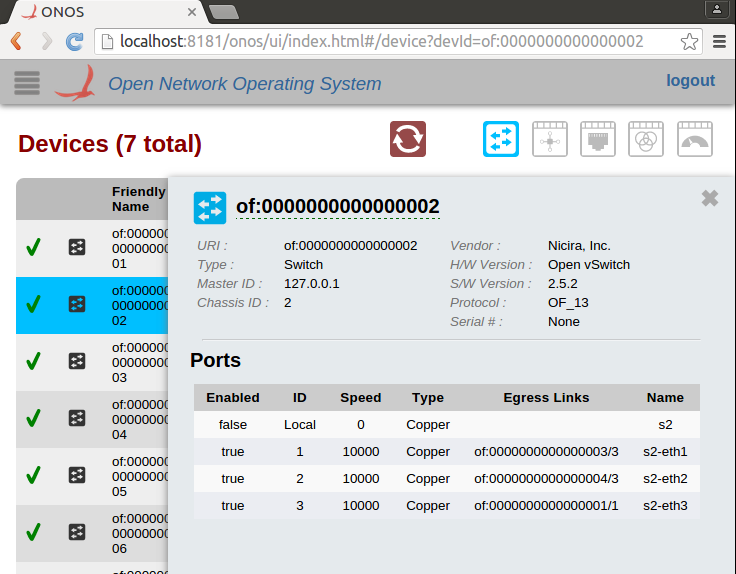


Chúng ta có thể kiểm tra các thông số của các thiết bị switch bằng cách kích chuột vào các thiết bị trên topo trên góc bên trái xuất hiện hình ảnh như thế này, kích vào các nút như trên ảnh được tô hình vuông dưới đây để kiểm tra thông số kĩ thuật của các thiết bị:

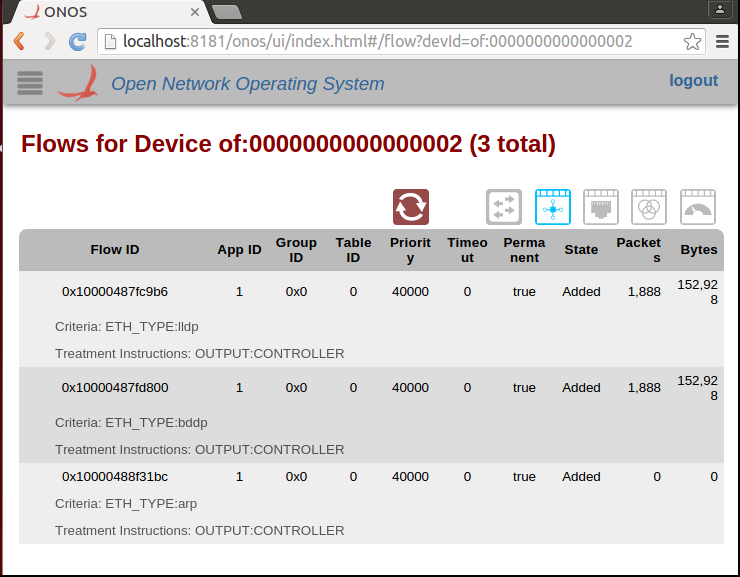


Chúng ta có thể kiểm tra thông tin, và có kết quả như sau:

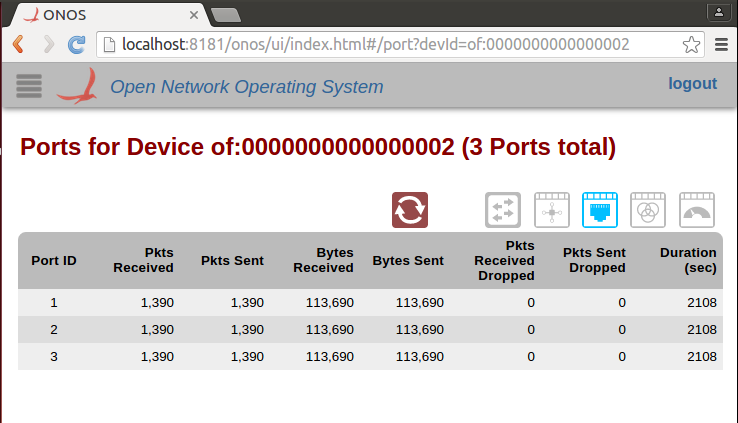
Hiện thiết bị:



Flow View:



Xem cổng thiết bị:



**Tài liệu tham khảo**

Trang Web: <https://www.opennetworking.org/sdn-resources/sdn-definition>

<https://wiki.onosproject.org/display/ONOS/Wiki+Home>