HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Nguyễn Thị Lan Hương

ỨNG DỤNG CÔNG NGHỆ SDN VÀO HỆ THỐNG MẠNG NỘI BỘ CỦA TRƯỜNG ĐẠI HỌC HÀ NỘI

LUẬN VĂN THẠC SĨ KỸ THUẬT

(***Theo định hướng ứng dụng***)

HÀ NỘI - NĂM 2019

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Nguyễn Thị Lan Hương

ỨNG DỤNG CÔNG NGHỆ SDN VÀO HỆ THỐNG MẠNG NỘI BỘ CỦA TRƯỜNG ĐẠI HỌC HÀ NỘI

Chuyên ngành: Kỹ thuật viễn thông

Mã số: 8.52.02.08

LUẬN VĂN THẠC SĨ KỸ THUẬT

(***Theo định hướng ứng dụng***)

NGƯỜI HƯỚNG DẪN KHOA HỌC : PGS.TS. NGUYỄN TIẾN BAN

HÀ NỘI - NĂM 2019

LỜI CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của riêng tôi.

Các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Người cam đoan

Nguyễn Thị Lan Hương

LỜI CẢM ƠN

Để thực hiện và hoàn thành luận văn này, em đã nhận được sự hỗ trợ, giúp đỡ cũng như là quan tâm, động viên từ các thầy cô nhà trường, cơ quan, và bạn bè. Luận văn cũng được hoàn thành dựa trên sự tham khảo, học tập kinh nghiệm từ các kết quả nghiên cứu liên quan, các sách, báo chuyên ngành của nhiều tác giả ở các trường Đại học, các tổ chức nghiên cứu, tổ chức chính trị…Đặc biệt hơn nữa là sự hướng dẫn của cán bộ giáo viên trường Học viện Công nghệ Bưu Chính Viễn Thông và sự giúp đỡ, tạo điều kiện về vật chất, tinh thần từ phía gia đình, bạn bè và các đồng nghiệp.

Trước hết, em xin gửi lời cảm ơn sâu sắc đến Thầy PGS.TS. Nguyễn Tiến Ban -người trực tiếp hướng dẫn khoa học đã luôn dành nhiều thời gian, công sức hướng dẫn em trong suốt quá trình thực hiện nghiên cứu và hoàn thành luận văn.

Em xin trân trọng cám ơn Ban giám hiệu Học viện Công nghệ Bưu Chính Viễn Thông cùng toàn thể các thầy cô giáo công tác trong trường đã tận tình truyền đạt những kiến thức quý báu, giúp đỡ em trong quá trình học tập và nghiên cứu.

Tuy có nhiều cố gắng, nhưng trong luận văn không tránh khỏi những thiếu sót. Em kính mong Quý thầy cô, các chuyên gia, những người quan tâm đến đề tài, đồng nghiệp, gia đình và bạn bè tiếp tục có những ý kiến đóng góp, giúp đỡ để đề tài được hoàn thiện hơn.

Một lần nữa em xin chân thành cám ơn!

Hà Nội, ngày 15 tháng 11 năm 2019

Học viên

Nguyễn Thị Lan Hương

MỤC LỤC

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

Viết tắt	Tiếng anh	Tiếng việt
AC	Access Controller	Bộ điều khiển truy cập
ACL	Access Control List	Danh sách điều khiển truy cập
AP	Access Point	Điểm truy cập
API	Application Programming Interface	Giao diện lập trình ứng dụng
ASIC	Application Specific Integrated Circuits	Ứng dụng mạch tích hợp cụ thể
BYOD	Bring Your Own Device	Mạng thiết bị của riêng bạn
CAPEX	Capital Expenditure	Chi phí vốn
CAPWAP	Control And Provisioning of Wireless Access Points	Kiểm soát và cung cấp các điểm truy cập không dây
FIB	Forwarding Information Base	Cơ sở thông tin chuyển tiếp
ICMP	Internetwork Control Message Protocol	Giao thức tin nhắn điều khiển mạng
NSF	Nonstop Forwarding	Chuyển tiếp không ngừng
ONF	Open Networking Foundation	Tổ chức mạng mở
OPEX	Operational Expenditure	Chi phí hoạt động
QoS	Quality of Service	Chất lượng dịch vụ
RIB	Routing Information Base	Cơ sở thông tin định tuyến
SDN	Software Defined Networking	Mạng điều khiển bởi phần phềm
SNMP	Simple Network Management Protocol	Giao thức quản lý mạng đơn giản
TCP/IP	Transmission Control Protocol/	Giao thức kiểm soát truyền/giao

	Internet Protocol	thức Internet
VLAN	Virtual Local Area Network	Mạng cục bộ ảo
VM	Virtual Machine	Máy ảo
VRF	Virtual Routing Forwarding	Chuyển tiếp định tuyến ảo
VSM	Virtual Supervisor Module	Module giám sát ảo

DANH MỤC BẢNG

DANH MỤC HÌNH VẼ

LỜI MỞ ĐẦU

Mạng Internet ra đời đã tạo nên một cuộc cách mạng trong công nghệ thông tin. Nó giúp mọi sự giao tiếp và trao đổi kiến thức, thông tin của con người trở nên dễ dàng hơn tạo nền tảng cho nền kinh tế tri thức hiện nay. Tuy nhiên, kiến trúc mạng truyền thống đã không hề có sự thay đổi trong hàng nửa thế kỷ qua và đang ngày càng trở nên không phù hợp với nhu cầu kinh doanh của các doanh nghiệp, các nhà khai thác mạng cũng như người dùng cuối. Hiện nay nhu cầu về nghiệp vụ ngày càng phức tạp của các doanh nghiệp và mức độ đa dạng về ứng dụng của các end-user đang ngày càng gia tăng, kéo theo đó là nhu cầu khác nhau của người dùng về mạng kết nối. Mạng cần phải đáp ứng việc thay đổi nhanh chóng các thông số về độ trễ, băng thông, định tuyến, bảo mật, … theo các yêu cầu của các ứng dụng. Chính vì thế rất nhiều chuyên gia đã đặt kỳ vọng vào một mô hình mạng mới, mạng điều khiển bởi phần mềm SDN.

Luận văn này cho chúng ta thấy một cách tổng quan về mạng SDN và giao thức OpenFlow cũng như quá trình áp dụng vào mô hình mạng nội bộ trường Đại học Hà Nội.

Luận văn gồm 3 chương:

Chương 1: Tổng quan về SDN.

Chương 2: Giao thức OpenFlow.

Chương 3: SDN trong mạng campus và ứng dụng vào mạng nội bộ trường Đại học Hà Nội.

Do kiến thức và thời gian có hạn, luận văn này không tránh khỏi sai sót, kính mong các thầy ,cô góp ý kiến để luận văn được hoàn thiện hơn.

CHƯƠNG 1: TỔNG QUAN VỀ SDN

Chính thức ra đời vào khoảng 2008 tại đại học Stanford, Mĩ nhưng Mạng khả trình (Software Defined Networking – SDN) đã tạo ra một cuộc cách mạng trong nền công nghiệp IT, và dự đoán rằng trong thời gian tới, SDN sẽ thay thế toàn bộ hệ thống mạng truyền thống. 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 client – server chiếm ưu thế, nhưng kiến trúc cố định như vậy không thích hợp với yêu cầu tính toán đa dạng, linh hoạt 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.

1.1. Đặt vấn đề

Bộ giao thức truyền thống TCP/IP được xem như là một chuẩn sử dụng từ giữa những năm 80 của thế kỷ trước. Đây là một hệ thống điều khiển cồng kềnh và không linh hoạt đối với mạng máy tính. Vì nó vừa " nghĩ" vừa "làm", điều đó có nghĩa là đầu tiên nó phải giải quyết bài toán xây dựng định tuyến, sau đó là áp dụng các tuyến đường này.

Trong các mạng hiện tại, chức năng điều khiển và truyền tải dữ liệu được kết hợp, đi liền với nhau, nó làm cho việc kiểm soát, điều khiển rất phức tạp. Cách tiếp cận dựa trên TCP/IP này gây ra một số hạn chế rất nghiêm trọng trong hoạt động với các tài nguyên của mạng.

Dễ thấy rằng số lượng và tính phức tạp của các giao thức rất lớn và phức tạp (Ngày nay số giao thức và các phiên bản giao thức được sử dụng thường xuyên đã vượt quá 600), việc kết hợp sự điều khiển và truyền dữ liệu làm cho quá trình kiểm soát cũng như điều khiển hoạt động mạng trở nên quá phức tạp đòi hỏi người quản

lý phải có tay nghề và chuyên môn cao. Vấn đề bảo mật đến thời điểm hiện tại vẫn không có giải pháp nào có độ tin cậy quá cao. Việc thêm vào bất kỳ sự thay đổi nào trong các thiết bị của mạng đều mất rất nhiều thời gian, chi phí cao và bắt buộc phải có sự tham gia của nhà sản xuất (vì tính độc quyền). Và vì thế, không ai có thể đảm bảo rằng những thiết bị mạng này chỉ chứa các chức năng đã được mô tả trong các tài liệu đính kèm sản phẩm. Đây là lí do vì sao có rất nhiều vụ bê bối nghe lén và đánh cắp dữ liệu diễn ra thời gian qua. Các thiết bị của mạng ngày nay là những thiết bị mang tính độc quyền, thiết bị "đóng", cản trở cho sự đổi mới, cập nhật và phát triển từ hướng người chủ của mạng, hay cộng đồng mạng.

Việc đáp ứng tất cả các nhu cầu hiện tại của thị trường gần như là không thể với mô hình mạng truyền thống. Phòng quản trị mạng của các công ty phải tìm cách hạn chế tối đa mạng của mình với việc sử dụng các công cụ điều khiển ở mức độ thiết bị và sử dụng các quá trình điều khiển bằng tay, lý do của vấn đề này chính là vì ngân sách được chi cho họ ngày càng bị cắt giảm, nếu may mắn thì chỉ được duy trì không đổi. Với những nhà khai thác mạng, họ cũng gặp vấn đề tương tự. Ta có thể thấy nhu cầu đối với tính di động và băng thông đang bùng phát (ngày nay số lượng người dung mạng máy tính trên kỹ thuật không dây vượt quá số người dùng mạng cố định, số lượng thiết bị di động trên đầu người ở các nước phát triển đã lớn hơn 3) trong khi đó lợi nhuận thu về ngày càng ít do các chi phí cho thiết bị và do việc giảm thu nhập. Các cấu trúc hiện tại của mạng không được tạo ra để thỏa mãn nhu cầu của người dùng hiện đại, của các công ty hay nhà khai thác mạng. Chúng ta sẽ phân tích một số giới hạn của mạng hiện tại, bao gồm:

- Tính phức tạp dẫn đến tình trạng trì trệ: Các kỹ thuật mạng ngày nay bao gồm các bộ giao thức rời rạc. Những giao thức này dùng để nối các host với nhau một cách tin cậy, với khoảng cách, tốc độ liên lạc, topo bất kỳ. Để thỏa mãn nhu cầu kinh doanh và yêu cầu kỹ thuật trong hơn chục năm trở lại đây, ngành công nghiệp đã phát triển các giao thức mạng dể hỗ trợ hiệu suất cũng như độ tin cậy cao hơn, có thể kết nối rộng hơn và độ bảo mật nghiêm ngặt hơn. Các giao thức này, về nguyên tắc, được tạo ra một cách cô lập, tuy nhiên mỗi giao thức giải quyết một vấn đề cụ

thể. Điều này dẫn đến một trong những hạn chế chính của mạng hiện tại đó là tính phức tạp. Ví dụ : để thêm vào hoặc dịch chuyển một thiết bị bất kỳ, người quản trị phải can thiệp đến một số thiết bị khác như : các bộ chuyển mạch, định tuyến, tường lửa… và phải cập nhật lại danh sách ACL (Access Control List), VLANs, QoS, và cả các cơ chế khác. Liên quan đến tính phức tạp này, các mạng hiện tại vì thế được xem như ở trạng thái "tĩnh" vì người quản trị phải cố gắng hạn chế đến mức thấp nhất những nguy cơ gián đoạn cung cấp dịch vụ. Tính "tĩnh" của mạng hiện tại lại là một mâu thuẫn rất lớn đối với đặc tính "động" của môi trường server ngày nay, ở đó việc ảo hóa các server làm tăng số lượng host một cách chóng mặt, đồng thời nó làm thay đổi quan điểm về vị trí vật lý của các host. Trước ảo hóa, các ứng dụng đều nằm trên một server và trao đổi traffic với các client. Ngày nay, các ứng dụng phân bố rời rạc trên một vài máy ảo (VM-Virtual Machine), những máy ảo này trao đổi các luồng dữ liệu với nhau. Các VM này "tái định cư" để làm tối ưu hóa và cân bằng tải trên server. Ngoài việc áp dụng kỹ thuật ảo hóa, nhiều công ty đã làm việc trên nền mạng hội tụ IP để truyền dữ liệu, thoại, video... Trong khi đó, mạng hiện tại hỗ trợ các mức độ khác nhau của QoS cho các ứng dụng khác nhau và cung cấp những tài nguyên này hoàn toàn bằng tay. Người quản trị cần phải cài đặt thiết bị của từng nhà cung cấp một cách riêng lẻ, và dĩ nhiên phải thiết lập các tham số như băng thông, QoS trên từng phiên làm việc cho mỗi ứng dụng. Do tính "tĩnh" của mình, mạng hiện tại không thể điều chỉnh một cách linh động so với những traffic luôn thay đổi của các ứng dụng và người dùng.

- Các chính sách không đồng nhất: Để thực hiện các chính sách mạng, người quản trị mạng cần phải cấu hình hàng ngàn thiết bị. Ví dụ mỗi lần áp dụng một máy ảo mới, phải tốn hàng giờ, thỉnh thoảng là hàng ngày để cấu hình lại các danh sách ACL trên toàn mạng. Tính phức tạp của mạng hiện tại làm cho công việc này trở nên khó khăn đối với các nhà quản trị để có thể áp dụng một bộ phối hợp truy cập, hay quy tắc bảo mật, QoS và các chính sách người dùng khác.

- Không có khả năng mở rộng: Vì các nhu cầu đối với các Data Center tăng nhanh chóng, nên mạng cũng buộc phải tăng (kích thước) theo. Tuy nhiên, mạng vì

thế quá phức tạp với hàng trăm, hàng ngàn thiết bị, những thiết bị này lại cần phải được cấu hình và điều khiển. Các nhà quản trị cũng buộc phải dựa trên các dự báo về traffic để mở rộng mạng. Nhưng trong các Trung tâm dữ liệu ảo hóa ngày nay, traffic là khác niệm "động" không tưởng và gần như không thể dự báo trước. Các nhà khai thác lớn như Google, Yahoo, Facebook… đã gặp phải các vấn đề phức tạp trong mở rộng kích thước mạng. Những nhà cung cấp dịch vụ này sử dụng các thuật toán xử lý song song ở quy mô lớn. Vì quy mô các ứng dụng đối với một người dùng cụ thể ngày càng tăng, số lượng các phần tử cần tính toán từ đó cũng tăng lên đến mức "bùng nổ" và các dữ liệu trao đổi giữa các node có thể đạt đến PB (Petabyte=1000 TB). Những công ty này cần phải đảm bảo hiệu suất cao, chi phí kết nối giữa hàng ngàn thiết bị thấp… Quy mô như vậy là không thể thực hiện với cách cấu hình bằng tay. Để duy trì khả năng cạnh tranh, các nhà khai thác cần phải thực hiện, cung cấp nhiều hơn các dịch vụ riêng biệt, khác biệt cho các client. Tính đa nhiệm cũng làm phức tạp bài toán hơn, vì mạng cần phục vụ nhiều nhóm người dùng với các ứng dụng khác nhau và các nhu cầu về hiệu suất khác nhau. Những nhà khai thác lớn, những nhà khai thác đóng vai trò chủ đạo trong quản lý traffic client rất khó để đáp ứng các nhu cầu với quy mô hiện tại của họ.

- Phụ thuộc vào nhà sản xuất: Các nhà mạng và các công ty cố gắng áp dụng các khả năng và dịch vụ mới trong việc đáp ứng các nhu cầu (những nhu cầu này thay đổi liên tục và rất nhanh) kinh doanh hoặc nhu cầu người dùng. Tuy nhiên khả năng của họ phụ thuộc vào các chu kỳ cập nhật firmware thiết bị của nhà sản xuất. Và điều đáng nói là những chu kỳ này có thể kéo dài lên đến 3 năm hoặc nhiều hơn nữa. Ngoài ra việc thiếu các chuẩn hóa, hay giao diện mở làm giới hạn khả năng điều chỉnh mạng của các nhà mạng. Chính sự không tương ứng giữa nhu cầu trên thị trường và khả năng của mạng đã dẫn đến "điểm gãy khúc". Đáp lại vấn đề này, mạng điều khiển bởi phần mềm SDN (Software-Defined Networking) đã được tạo ra.

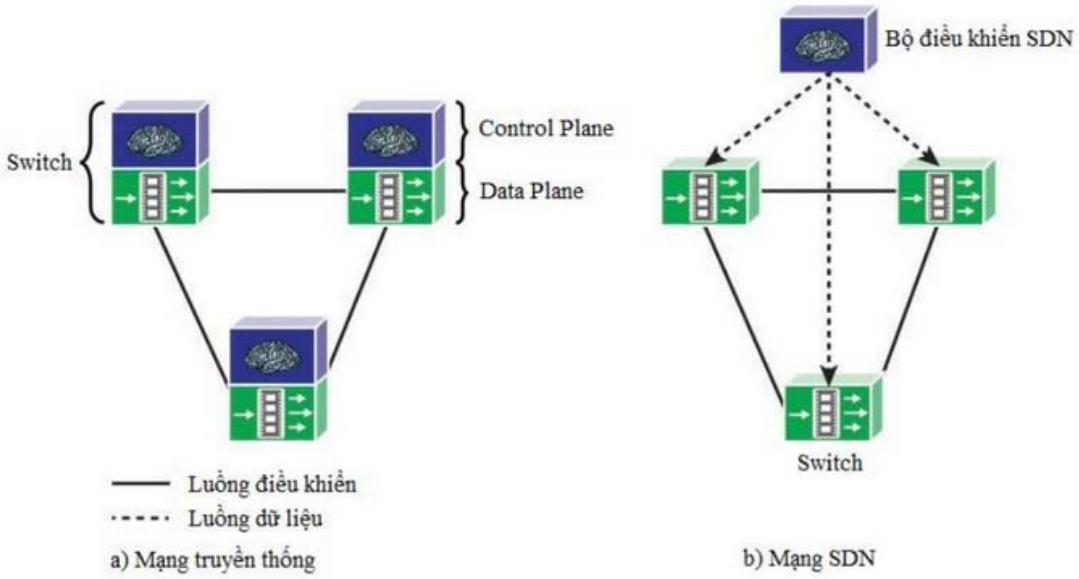
1.2. Khái niệm và cấu trúc mạng SDN

1.2.1. Khái niệm về SDN

Trước khi đưa ra khái niệm về SDN, ta thử đặt ra một giả thiết là nếu ta có thể tách rời phần điều khiển ra khỏi các thiết bị mạng thì điều đó có thể làm cho khả năng xử lý của thiết bị tăng lên hay không? Có thể tạo ra một mạng thông minh hơn và linh hoạt hơn hay không?

Thực tế là dựa trên giả thiết đó, người ta đã nghiên cứu và phát triển thành một mạng mà ở đó nhiệm vụ điều khiển mạng được xử lý bởi các bộ điều khiển và các bộ điều khiển đó có thể thao tác tới phần cứng, bộ nhớ và các chức năng của các thiết bị router, switch để đạt được mục đích của người sử dụng. Do đó, mạng trở nên linh hoạt hơn, hiệu suất sử dụng cao hơn và dễ quản lý hơn bao giờ hết.

Để hiểu rõ hơn ta xem xét sự khác nhau giữa chức năng của các thiết bị của mạng truyền thống và mạng SDN.



Hình 1.1: So sánh kiến trúc mạng truyền thống và SDN

Hình 1.1 (a) mô tả sơ đồ một mạng truyền thống đơn giản. Đối với mạng truyền thống thì các thiết bị định tuyến hoặc chuyển mạch trao đổi các thông tin với nhau và quá trình tính toán xử lý đều xảy ra ở mỗi node mạng (ở tại mỗi

router/switch). Chức năng chính của các thiết bị mạng như router/switch là vận chuyển dữ liệu, như ta thấy ở mô hình trên thì các thiết bị không được hoàn toàn tập trung vào chức năng đó. Nhưng đối với mạng SDN thì điều đó lại là khác.

Hình 1.1 (b) mô tả sơ đồ mạng đơn giản với bộ điều khiển SDN. Theo như hình 1.1 (b) thì ta thấy việc thu thập thông tin của các thiết bị trong mạng và tính toán xử lý các thông tin thu thập được đều được chuyển đến một bộ điều khiển mạng (Bộ điều khiển SDN). Các thiết bị router/switch chỉ tập trung vào chức năng vận chuyển dữ liệu. Điều đó làm cho việc quản lý mạng trở nên đơn giản hơn và các thiết bị phần cứng có thể nâng công suất làm việc lên.

Từ sự so sánh trên ta rút ra được một số điểm khác nhau giữa 2 mạng đó là:

♣ Phần điều khiển và phần vận chuyển dữ liệu:

- Mạng truyền thống: Đều được tích hợp trong thiết bị mạng.

- Mạng SDN: Phần điều khiển được tách riêng khỏi thiết bị mạng và được chuyển đến một thiết bị được gọi là bộ điều khiển SDN.

♣ Phần thu thập và xử lý các thông tin:

- Mạng truyền thống: Được thực hiện ở tất cả các phần tử trong mạng.

- Mạng SDN: Được tập trung xử lý ở bộ điều khiển SDN.

♣ Khả năng lập trình bởi các ứng dụng:

- Mạng truyền thống: mạng không thể được lập trình bởi các ứng dụng. Các thiết bị mạng phải được cấu hình một cách riêng lẽ và thủ công.

- Mạng SDN: Mạng có thể lập trình bởi các ứng dụng, bộ điều khiển SDN có, thể tương tác đến tất cả các thiết bị trong mạng.

Hiện nay có rất nhiều định nghĩa về mạng SDN nhưng theo ONF (Open Networking Foundation - một tổ chức phi lợi nhuận đang hỗ trợ việc phát triển SDN thông qua việc nghiên cứu các tiêu chuẩn mở phù hợp) thì mạng SDN được định nghĩa như sau: “ Sofware Defined Network (SDN) là một kiểu kiến trúc mạng mới, động, dễ quản lý, chi phí hiệu quả, dễ thích nghi và rất phù hợp với nhu cầu mạng ngày càng tăng hiện nay. Kiến trúc này phân tách phần điều khiển mạng (Control Plane) và chức năng vận chuyển dữ liệu (Forwarding Plane or Mặt bằng dữ liệu),

điều này cho phép việc điều khiển mạng trở nên có thể lập trình và cơ sở hạ tầng mạng độc lập với các ứng dụng và dịch vụ mạng". Phần điều khiển được tách rời và được tập trung ở bộ điều khiển SDN. Điều này có nghĩa là các thiết bị mạng ở lớp thiết bị phần cứng không cần phải hiểu và xử lý các giao thức phức tạp mà chúng chỉ chấp nhận và vận chuyển dữ liệu theo một con đường nào đó dưới sự chỉ huy của bộ điều khiển SDN.

Dựa vào bộ điều khiển SDN mà các nhà khai thác mạng và quản trị mạng có thể lập trình cấu hình trên đó thay vì phải thực hiện thủ công hàng ngàn câu lệnh cấu hình trên các thiết bị riêng lẽ. Ngoài ra nó còn có thể triển khai các ứng dụng mới và các dịch vụ mạng một cách nhanh chóng.

1.2.2. Cấu trúc của mạng SDN

Để có cái nhìn tổng quan hơn ta xem xét đến kiến trúc của SDN. Kiến trúc này tách biệt hai cơ chế đang tồn tại trong kiến trúc mạng hiện tại là cơ chế điều khiển và cơ chế chuyển tiếp. Các đặc tính trong kiến trúc SDN:

- Khả năng lập trình trực tiếp: Việc điều khiển mạng được lập trình trực tiếp bởi nó đã được tách biệt với các chức năng chuyển tiếp.

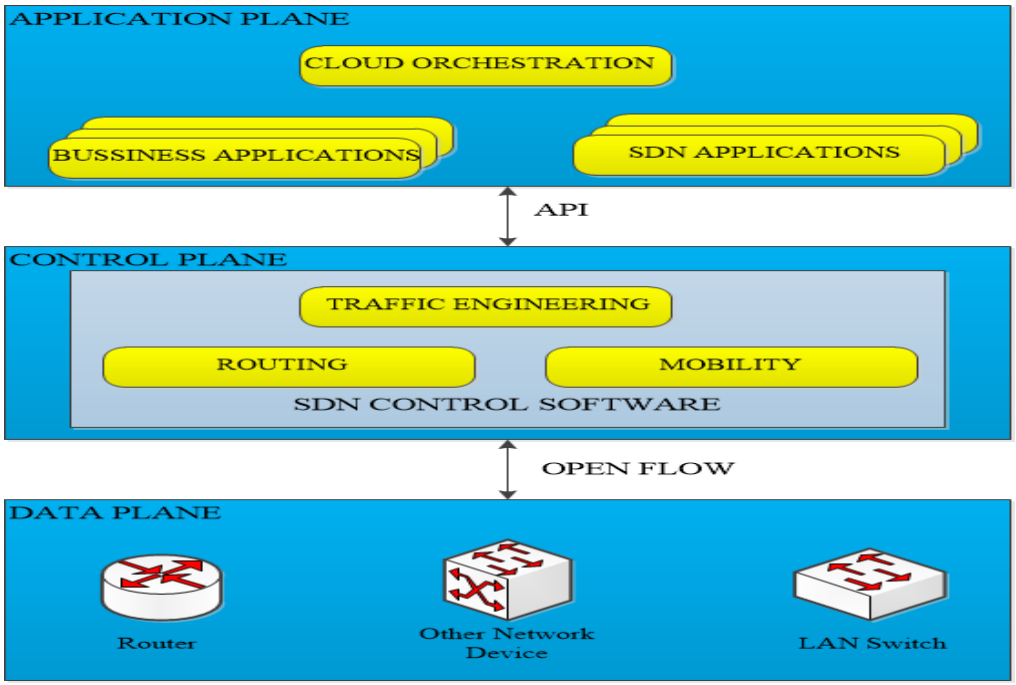
- Nhanh chóng: Việc tách biệt các chức năng điều khiển và chức năng chuyển tiếp cho phép các nhà quản trị mạng linh hoạt trong việc điều chỉnh luồng lưu lượng của mạng khi có yêu cầu thay đổi.

- Quản lý tập trung: Việc điều khiển tập trung được thực hiện bởi bộ điều khiển SDN (một phần mềm), cho ta thấy được cái nhìn tổng quan về mạng.

- Việc cấu hình lập trình được: SDN cho phép người quản trị mạng cấu hình, quản lý, thiết lập bảo mật, tối ưu hóa tài nguyên mạng nhanh chóng nhờ có các chương trình hỗ trợ SDN đã tự động hóa, những chương trình đó hoàn toàn có thể tự lập trình được mà không phụ thuộc vào phần mềm.

- Cung cấp các tiêu chuẩn mở: Khi triển khai thông qua các tiêu chuân mở, SDN đã đơn giản hóa việc thiết kế mạng và vận hành bởi vì các chỉ dẫn được cung cấp bởi bộ điều khiển SDN thay vì các giao thức hay các thiết bị chuyên biệt của các nhà cung cấp.

Kiến trúc của SDN gồm 3 lớp riêng biệt: lớp ứng dụng, lớp điều khiển, và lớp cơ sở hạ tầng (lớp chuyển tiếp). Mô hình kiến trúc mạng SDN được thể hiện ở hình 1.2.



Hình 1.2: Kiến trúc của mạng SDN

- Lớp ứng dụng (Application Plane): bao gồm các ứng dụng được triển khai trên mạng, kết nối với lớp điều khiển thông qua các API (Application Programming Interface – Giao diện lập trình ứng dụng), cho phép lớp ứng dụng lập trình lại (cấu hình lại) mạng (Ví dụ: điều chỉnh tham số độ trễ, băng thông, định tuyến, …) thông qua lớp điều khiển lập trình giúp cho hệ thống mạng tối ưu hoạt động theo một yêu cầu nhất định.

- Lớp điều khiển (Control Plane): là nơi tập trung các bộ điều khiển SDN thực hiện việc điều khiển cấu hình mạng theo các yêu cầu từ lớp ứng dụng và khả năng của mạng. Các bộ điều khiển này có thể là các phần mềm được lập trình trực tiếp.

Các bộ điều khiển SDN xác định các luồng dữ liệu sẽ đi qua lớp dữ liệu phía dưới và mỗi luồng dữ liệu đi qua mạng đều phải có sự cho phép của bộ điều khiển SDN và khi được sự cho phép thì bộ điều khiển sẽ tính toán chọn đường đi tốt nhất cho các luồng dữ liệu. Một bộ điều khiển là một ứng dụng quản lý kiểm soát luồng lưu lượng trong môi trường mạng. Để truyền thông tin điều khiển đến lớp cơ sở hạ tầng, lớp điều khiển sử dụng các giao thức như: OpenFlow, ONOS, ForCES, PCEP, NETCONF, SNMP hoặc thông qua các cơ chế riêng biệt. Hầu hết các bộ điều khiển SDN hiện nay dựa trên giao thức OpenFlow. Giao thức này sẽ được đề cập ở chương sau.

Bộ điều khiển SDN hoạt động như một loại hệ điều hành cho mạng. Tất cả thông tin liên lạc giữa các ứng dụng và các thiết bị phải đi qua bộ điều khiển. Cùng với chức năng chính, nó có thể tiếp tục được mở rộng để thực hiện thêm các nhiệm vụ quan trọng như định tuyến và truy cập mạng.

Tóm lại, vai trò của lớp điều khiển là: cung cấp API để có thể xây dựng các ứng dụng cho hệ thống mạng, thu nhận thông tin từ hệ thống mạng vật lý, điều khiển hệ thống mạng vật lý.

- Lớp cơ sở hạ tầng (Infrastructure/ Data Plane): bao gồm các thiết bị mạng thực tế (vật lý hoặc 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.

Kiến trúc SDN rất linh hoạt, nó có thể hoạt động với các loại thiết bị chuyển mạch và các giao thức khác nhau. Trong kiến trúc SDN, thiết bị chuyển mạch thực hiện các chức năng sau:

+ Đóng gói và chuyển tiếp các gói tin đầu tiên đến bộ điều khiển SDN để bộ điền khiển SDN quyết định các flow entry sẽ được thêm vào flow table của switch.

+ Chuyển tiếp các gói tin đến các cổng thích hợp dựa trên flow table.

+ Flow table có thể bao gồm các thông tin ưu tiên được quyết định bởi bộ điều khiển SDN.

+ Switch có thể hủy các gói tin trên một luồng riêng một cách tạm thời hoặc vĩnh viễn nhưng dưới sự cho phép của bộ điều khiển.

Bộ điều khiển SDN quản lý các trạng thái chuyển tiếp của các thiết bị switch trong mạng, việc quản lý này được thông qua một bộ giao diện mở API, nó cho phép bộ điều khiển SDN có thể giải quyết các yêu cầu mà không cần thay đổi các thành phần cấp dưới của mạng, bao gồm cả mô hình mạng. Với sự tách riêng biệt miền điều khiển và miền dữ liệu, SDN cho phép các ứng dụng triển khai một cách dễ dàng mà không cần quan tâm chi tiết đến việc hoạt động của các thiết bị mạng.

1.3. Ưu nhược điểm của SDN so với mạng IP

Mạng truyền thống và mạng SDN đều có những ưu nhược điểm riêng nhưng với những thuộc tính quan trọng ví dụ như dễ quản lý hơn cho người quản trị, chi phí và độ phức tạp giảm thì người ta đánh giá rằng mạng SDN phù hợp hơn so với mạng truyền thống. Những lợi ích mà mạng SDN đem lại gồm:

- Quản lý tập trung và đơn giản: dựa vào SDN mà người quản trị mạng có thể có quyền kiểm soát mạng một cách đơn giản và hiệu quả mà không cần có quyền truy cập trực tiếp đến phần cứng thiết bị. Họ chỉ cần thông qua các API đã được cung cấp để có thể xây dựng ứng dụng cho toàn hệ thống mạng.

- Truyền tải nhanh chóng và linh hoạt: SDN cung cấp một cơ chế điều khiển duy nhất đối với cơ sở hạ tầng mạng và giảm bớt sự phức tạp của các quá trình xử lý thông qua sự tự động hóa., giúp các doanh nghiệp triển khai nhanh hơn các ứng dụng, các dịch vụ và cơ sở hạ tầng mạng.

- Cho phép thay đổi: SDN cho phép sử dụng không hạn chế và có thể thay đổi các chính sách mạng để phát hiện sự xâm nhập, tường lửa và tạo sự cân bằng với sự thay đổi của phần mềm.

- Giảm CapEx (chi phí đầu tư): SDN giúp giảm thiểu các yêu cầu mua phần cứng theo mục đích xây dựng các dịch vụ, phần cứng mạng, loại bỏ lãng phí cho việc dự phòng.

- Giảm OpEx (chi phí vận hành): nhờ khả năng lập trình được các phần tử mạng, SDN giúp dễ dàng thiết kế, triển khai, quản lý và mở rộng mạng. Khả năng phối hợp và dự phòng tự động không những giảm thời gian quản lý tổng thể, mà còn giảm xác suất lỗi do con người hướng tới việc tối ưu khả năng và độ tin cậy của dịch vụ.

- Mở ra cơ hội cho các nhà cung cấp thiết bị trung gian khi phần điều khiển được tách rời khỏi phần cứng. Với SDN, việc điều khiển được tập trung tại bộ điều khiển SDN, các thiết bị mạng chỉ có nhiệm vụ chuyển tiếp gói tin do đó sự khác biệt giữa những nhà sản xuất không ảnh hưởng tới toàn hệ thống mạng.

Bên cạnh những ưu điểm đã có của mình, mạng SDN vẫn tồn tại một số nhược điểm sau:

- Đầu tiên là vấn đề bảo mật, nếu SDN bị tấn công thành công vào hệ thống điều khiển thì mạng có thể được truy cập và thiết lập các thay đổi từ bất cứ đâu, bất cứ thời điểm nào. Đối với mạng truyền thống điều này khó có thể xảy ra vì để có thể truy cập vào mạng ta phải có quyền truy cập vào phần cứng của thiết bị. Hầu hết các công ty và doanh nghiệp chỉ cho phép một số cá nhân thực hiện được việc đó nên hệ thống sẽ an toàn hơn.

- Tiếp theo, SDN là một kiến trúc mạng kiểu mới, các giao thức tương tác giữa các bộ điều khiển với nhau chưa được hoàn thiện một cách toàn diện nên việc phát triển mạng SDN vẫn còn hạn chế.

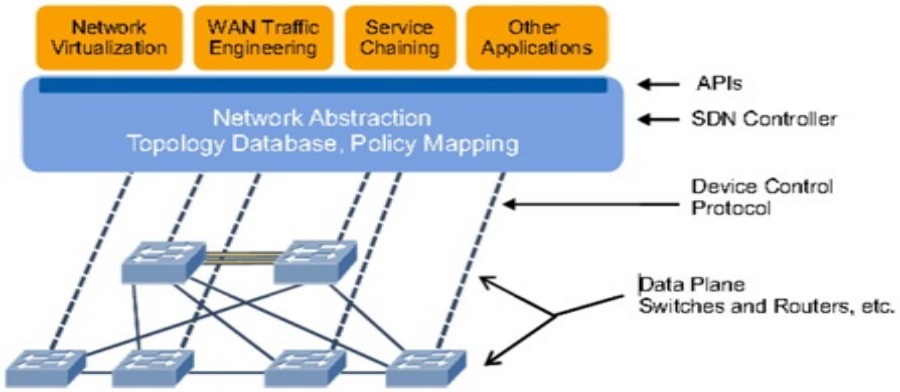
- Một vấn đề nữa là quá trình triển khai mạng SDN không thể hoàn thiện trong thời gian ngắn mà phải theo từng bước một. Các công ty, doanh nghiệp không thể trong một lúc thay thế toàn bộ các thiết bị hiện có thành OpenFlow Switch được bởi nó rất tốn kém.

Tóm lại, với những ưu nhược điểm của mạng SDN so với mạng truyền thống thì chúng ta vẫn có thể tin rằng đây là một kiến trúc mạng mới tốt hơn, linh hoạt hơn, đáp ứng được phần lớn nhu cầu của ứng dụng hiện nay.

1.4. Các mô hình triển khai mạng SDN

1.4.1. Switch Based

Ý tưởng của mô hình dựa trên switch là các giao thức điều khiển SDN được đưa ra trực tiếp từ bộ điều khiển SDN (tại máy ảo) đến lớp điều khiển, lớp cơ sở hạ tầng/ dữ liệu với các SDN switch.



Hình 1.3: Mô hình Switch Based

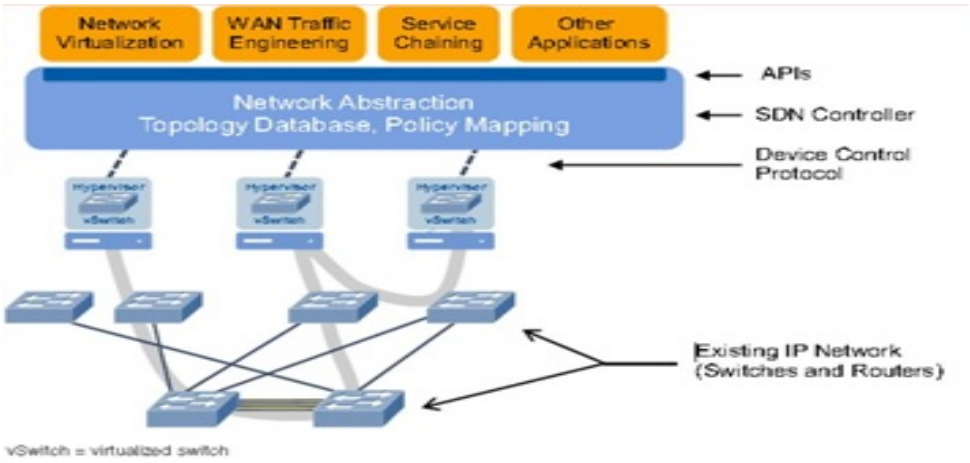
Đối với mạng thông thường khi một gói tin đến một switch, dựa vào các giao thức được xây dựng sẵn trong switch nó sẽ biết được nơi sẽ chuyển tiếp gói tin đến. Switch sẽ gửi các gói tin đi đến cùng một địa điểm, cùng một con đường và nó đối xử với các gói tin là như nhau. Trong các doanh nghiệp, các switch thông minh được thiết kế với các bảng mạch tích hợp ứng dụng cụ thể ASIC (Application Specific Integrated Circuits) điều đó giúp cho các switch nhận biết được các loại gói tin và xử lý chúng một cách thích hợp. Với việc có thêm các bảng mạch tích hợp ASIC làm cho các thiết bị switch trở nên đắt hơn so với các thiết bị chuyển mạch thông thường.

Đối với mạng SDN được mô tả như hình 1.3 thì người quản trị mạng có thể quản lý các lưu lượng dữ liệu từ một thiết bị kiểm soát trung tâm mà không cần phải tác động trực tiếp vào từng switch. Người quản trị mạng có thể thay đổi bất cứ các quy tắc chuyển mạch nào nếu cần thiết như ưu tiên, không ưu tiên hoặc thậm chí có thể chặn một số gói tin đặc thù nào đó. Điều này đặc biệt hữu ích cho kiến trúc đám mây bởi vì nó cho phép người quản trị có thể quản lý các luồng dữ liệu một cách linh hoạt và hiệu quả hơn. Kiến trúc này còn cho phép các kỹ sư mạng hỗ trợ đa kết nối qua các thiết bị phần cứng của nhiều nhà cung cấp khác nhau.

Điểm hạn chế lớn nhất của phương pháp này là không thể tận dùng tất cả các thiết bị mạng lớp 3 và lớp 2 của mạng truyền thống.

1.4.2. Overlay Network

Mô hình Overlay Network có thể tận dụng các thiết bị của mạng TCP/IP hiện có bằng cách ảo hóa. Ở mô hình này, các giao thức điều khiển của SDN đi trực tiếp từ bộ điều khiển SDN (tại máy ảo) đến các thiết bị chuyển mạch ảo (Hypervisor switch) để kiểm soát các thiết bị mạng ở lớp dưới.



Hình 1.4: Mô hình Overlay Network

Mô hình như hình 1.4 yêu cầu sử dụng các thiết bị chuyển mạch ảo để đáp ứng các lệnh đến các thiết bị mạng IP. Các thiết bị chuyển mạch ảo là các máy ảo

chịu trách nhiệm giao tiếp giữa thiết bị mạng IP với các máy ảo và các ứng dụng mạng SDN. Mô hình dùng chuyển mạch ảo có 2 chức năng đó là chức năng vận chuyển của lớp 2 thông qua một mô đun Ethernet ảo VEM (Virtual Ethernet Module) và tuân thủ các chính sách giám sát.

VEM cung cấp thông tin cấu hình, hỗ trợ chuyển mạch lớp 2 và hỗ trợ các chức năng nâng cao của mạng như cấu hình cho các cổng , chất lượng dịch vụ, bảo mật cho các cổng , VLAN và điều khiển truy cập. Ngoài ra, khi mất liên lạc với các thiết bị chuyển mạch ảo, mô đun Ethernet ảo có hỗ trợ chức năng Nonstop Forwarding (NSF) để có thể tiếp tục chuyển tiếp lưu lượng dựa trên cấu hình cuối cùng mà các bộ chuyển mạch được biết. Như vậy, mô đun Ethernet ảo cung cấp khả năng chuyển mạch với độ tin cậy cao cho môi trường máy chủ ảo.

Để kiểm soát nhiều mô đun Ethernet ảo người ta sử dụng một bộ giám sát ảo (Virtual Supervisor Module - VSM). Thay vì sử dụng nhiều thẻ chức năng vật lý, VSM hỗ trợ chạy nhiều VEM bên trong một máy chủ vật lý. Cấu hình được thực hiện thông qua VSM và tự động chuyển đến các VEM. Thay vì cấu hình các chuyển mạch mềm bên trong các hypervisor trên cơ sở các máy chủ với nhau, người quản trị có thể cấu hình ngay lập tức trên tất cả các VEM được quản lý bởi VSM từ một giao diện duy nhất. VSM còn cung cấp chức năng cấu hình các cổng thông qua phần mềm.

Mô hình triển khai này có ưu điểm là sử dụng được cơ sở hạ tầng mạng IP hiện tại nhưng nó cũng sẽ gây khó khăn cho các nhà quản trị vì phải duy trì hệ thống cũ và sửa chữa các vấn đề về định tuyến trong mạng SDN.

1.4.3. Mạng lai

Mô hình này là sự kết hợp giữa 2 mô hình Switch Based và Overlay Network. Cách triển khai mô hình này tận dụng được mạng lưới IP hiện có và dần dần loại bỏ mạng lưới cũ để chuyển sang sử dụng hoàn toàn các switch SDN. Điều này cho phép các doanh nghiệp có thể kiểm soát tốc độ triển khai SDN và chi phí đầu tư thiết bị mạng.

1.5. Ứng dụng của SDN

Với những lợi ích mà mình đem lại, SDN có thể triển khai trong phạm vi các doanh nghiệp hoặc trong các nhà cung cấp hạ tầng và dịch vụ viễn thông để giải quyết các yêu cầu của các nhà cung cấp tại mỗi phân khúc thị trường.

1.5.1. Phạm vi doanh nghiệp

1.5.1.1. Áp dụng trong mạng doanh nghiệp

Mô hình tập trung, điều khiển và dự phòng tự động của SDN hỗ trợ việc hội tụ dữ liệu, voice, video, cũng như là việc truy cập tại bất kỳ thời điểm nào, bất kỳ nơi đâu. Điều này được thực hiện thông qua việc cho phép các nhà quản trị mạng thực thi chính sách nhất quán trên cả cơ sở hạ tầng không dây lẫn có dây. Ngoài ra, SDN hỗ trợ việc quản lý và giám sát tự động tài nguyên mạng, xác định bằng các hồ sơ cá nhân và các yêu cầu ứng dụng, để đảm bảo tối ưu trải nghiệm người dùng với khả năng của nhà mạng.

1.5.1.2. Áp dụng trong các trung tâm dữ liệu (Data Center)

Việc ảo hóa các thực thể mạng của kiến trúc SDN cho phép việc mở rộng trong các trung tâm dữ liệu, dịch chuyển tự động các máy ảo, tích hợp chặt chẽ hơn với kho lưu trữ, sử dụng server tốt hơn, sử dụng năng lượng thấp hơn và tối ưu được băng thông hơn.

1.5.1.3. Áp dụng với dịch vụ điện toán đám mây (cloud)

Mặc dù được sử dụng để hỗ trợ cho điện toán đám mây riêng hay môi trường điện toán đám mây lai, SDN cho phép tài nguyên mạng được phân bố một cách linh hoạt, điều đó cho phép sự đáp ứng nhanh chóng của các dịch vụ điện toán đám mây và tạo sự chuyển giao linh hoạt hơn đến cho các nhà cung cấp điện toán đám mây bên ngoài. Với các công cụ an toàn để quản lý mạng ảo của họ, các doanh nghiệp và các đơn vị kinh doanh sẽ tin tưởng vào các dịch vụ đám mây nhiều hơn nữa.

1.5.2. Phạm vi các nhà cung cấp hạ tầng và dịch vụ viễn thông

SDN cung cấp cho các nhà mạng, các nhà cung cấp đám mây công cộng và các nhà cung cấp dịch vụ, sự mở rộng và tự động thiết kế để triển khai một mô hình tính toán có ích cho ItaaaS (IT as a Service). Điều này thực hiện thông qua việc đơn giản hóa triển khai các dịch vụ tùy chọn và theo yêu cầu, cùng với việc chuyển dời sang mô hình selfservice. Mô hình tập trung, dự phòng và điều khiển tự động của SDN dễ dàng hỗ trợ cho thuê linh hoạt các tài nguyên, đảm bảo tài nguyên mạng được triển khai ở mức tối ưu, giảm CAPEX và OPEX và tăng giá trí và tốc độ dịch vụ.

1.6. Kết luận chương

Với xu hướng người dùng di động, ảo hóa máy chủ và các dịch vụ ngày càng tăng dẫn đến kiến trúc mạng thông thường ngày nay không để đáp ứng xử lý kịp. Mạng SDN cho chúng ta một cái nhìn mới, khái niệm mới về một kiến trúc mạng động, dễ thích nghi, mở rộng và đáp ứng các dịch vụ phong phú. Với việc tách phần điều khiển và dữ liệu, kiến trúc mạng SDN cho phép mạng có thể lập trình và quản lý một cách dễ dàng hơn. SDN hứa hẹn sẽ chuyển đổi mạng lưới tĩnh ngày nay trở nên linh hoạt hơn với nền tảng có thể lập trình với sự thông minh để có thể tự động xử lý các hành vi một cách tự động. Với nhiều lợi thế của mình và động lực phát triển cao kiến trúc SDN đang trên đường để trở thành một tiêu chuẩn mới cho các mạng.

CHƯƠNG 2: GIAO THỨC OPENFLOW

Ý tưởng SDN đã bắt đầu được gần 10 năm, nhưng gần đây SDN mới bắt đầu được thực hiện bởi các công ty như Cisco Systems và Juniper Networks. Tuy nhiên các nhà sản xuất và khai thác mạng đã và đang bắt đầu làm quen với OpenFlow, một công nghệ hứa hẹn sẽ mang đến khả năng tương tác và hiệu suất hoạt động cao cho SDN. Với sự giúp đỡ của OpenFlow controller, các nhà quản trị mạng có thể xác định các cách thức và tuyến đường để truyền dữ liệu, thiết lập các quy tắc ưu tiên cho việc xử lý các gói tin và chuyển hướng dữ liệu qua các thiết bị chuyển mạch của mạng nội bộ hay mạng toàn cầu.

Chương này sẽ giới thiệu một cách tổng quan cho chúng ta biết về giao thức OpenFlow cũng như các thức hoạt động và các lợi ích mà nó mang lại.

2.1. Lịch sử và sự phát triển của OpenFlow

Vào tháng 3 năm 2011, các công ty Cisco, Facebook, Google, Microsoft…và nhiều công ty khác đã thành lập nên tổ chức Open Networking Foundation (ONF) để thúc đẩy công nghệ OpenFlow và giao thức chuyển mạch OpenFlow Switching Protocol. Tuy nhiên , một số chuyên gia cho rằng OpenFlow không có đủ khả năng để triển khai trên diện rộng và các nhà sản xuất có thể thêm vào công nghệ này các phần mở rộng độc quyền của mình, điều này làm mất đi khả năng tương tác vốn có của OpenFlow.

Ngày nay với sự phát triển nhanh chóng và rộng khắp của điện toán đám mây đã kích thích các nhu cầu về tính linh hoạt, độ tin cậy, an toàn và cần được quản lý tốt của mạng xương sống. Để giải quyết vấn đề này, cần phải có các hệ thống điều khiển thông minh hơn, hiệu quả hơn, những hệ thống cho phép phối hợp hoạt động của hàng ngàn thiết bị định tuyến và chuyển mạch. Hiện nay những thiết bị này chỉ cung cấp cho người sử dụng các khả năng tái lập trình một cách hạn chế, và để nâng cao tính hiệu quả ở các trung tâm xử lý dữ liệu (Data Center), những người quản trị hệ thống cần một sự kiểm soát chi tiết hơn, và khả năng mở rộng cao

hơn. Trong khi đó, mỗi nhà cung cấp có các bộ API và chức năng của riêng mình, điều này hạn chế khả năng điều khiển lưu lượng giữa các thiết bị của nhà sản xuất khác nhau.

Các thiết bị chuyển mạch truyền thống vừa thực hiện chuyển tiếp gói dữ liệu nhanh chóng (data path) vừa thực hiện định tuyến cấp cao (control path).Trong khi đó OpenFlow cung cấp chức năng điều khiển cao cấp độc lập với phần cứng, do đó đẩy nhanh quá trình chuyển tiếp và định tuyến. Ngoài ra, trong mạng OpenFlow, tất cả "phần thông minh" được đưa lên trên một server trung tâm, vì vậy nó thực hiện các hoạt động phức tạp một cách đơn giản hơn.

Các nhà nghiên cứu từ đại học Stanford và đại học California bắt đầu phát triển SDN vào năm 2002, còn dự án OpenFlow được bắt đầu trong năm 2008. Juniper và các nhà cung cấp khác sản xuất các sản phẩm SDN độc quyền trên cơ sở của OpenFlow từ năm 2010. OpenFlow đã giúp cho SDN trở nên khả thi hơn.

2.2. Giao thức OpenFlow

Để tách biệt hẳn phần điều khiển ra khỏi phần chuyển tiếp và cung cấp khả năng lập trình cho lớp điều khiển, ONF sử dụng giao thức OpenFlow. OpenFlow là tiêu chuẩn đầu tiên, cung cấp khả năng giao tiếp giữa các giao diện của lớp điều khiển và lớp chuyển tiếp trong kiến trúc SDN. OpenFlow cho phép truy cập trực tiếp và điều khiển lớp chuyển tiếp của các thiết bị mạng như switch và router, cả thiết bị vật lý lẫn thiết bị ảo.

- Các đặc trưng của OpenFlow:

+ OpenFlow có thể được sử dụng bởi ứng dụng phần mềm ngoài để điều khiển lớp chuyển tiếp của các thiết bị mạng, giống như tập lệnh của CPU điều khiển một hệ thống máy tính.

+ Giao thức OpenFlow được triển khai trên cả hai giao diện của kết nối giữa các thiết bị cơ sở hạ tầng mạng và phần mềm điều khiển SDN.

+ OpenFlow sử dụng khái niệm “flow” (luồng) để nhận dạng lưu lượng mạng trên cơ sở định nghĩa trước các quy tắc phù hợp (được lập trình sẵn bởi phần mềm điều khiển SDN). Giao thức này cũng cho phép định nghĩa cách mà lưu lượng phải được truyền qua các thiết bị mạng trên cơ sở các tham số, chẳng hạn như mô hình lưu lượng sử dụng, ứng dụng và tài nguyên đám mây. Do đó OpenFlow cho phép mạng được lập trình dựa trên cơ sở luồng lưu lượng. Một kiến trúc SDN trên cơ sở OpenFlow cung cấp điều khiển ở mức cực kỳ chi tiết, cho phép mạng phản hồi sự thay đổi theo thời gian thực của ứng dụng, người dùng và mức phiên. Mạng định tuyến trên cơ sở IP hiện tại không cung cấp mức này của điều khiển, tất cả các luồng lưu lượng giữa hai điểm cuối phải theo cùng một đường thông qua mạng, mặc dù yêu cầu của chúng khác nhau.

+ Giao thức OpenFlow là một chìa khóa để cho phép các mạng định nghĩa bằng phần mềm và cũng là giao thức tiêu chuẩn SDN duy nhất cho phép điều khiển lớp chuyển tiếp của các thiết bị mạng.

- Công nghệ SDN trên cơ sở OpenFlow cho phép các nhà quản trị giải quyết các ứng dụng băng thông cao và linh hoạt ngày nay, khiến cho mạng thích ứng với các nhu cầu kinh doanh thay đổi và giảm đáng kể các hoạt động quản lý phức tạp. Những lợi ích mà các doanh nghiệp và nhà khai thác mạng có thể đạt được thông qua kiến trúc SDN trên cơ sở OpenFlow bao gồm:

+ Tập trung hóa điều khiển trong môi trường nhiều nhà cung cấp thiết bị: dựa trên giao thức OpenFlow phần mềm điều khiển SDN có thể điều khiển bất kỳ thiết bị mạng nào của bất kỳ nhà cung cấp thiết bị nào bao gồm switch, router và các switch ảo.

+ Giảm sự phức tạp thông qua việc tự động hóa: kiến trúc SDN trên cơ sở OpenFlow cung cấp chức năng quản lý mạng tự động và linh hoạt bằng việc phát triển các phần mềm chạy trên các bộ điều khiển.

+ Tốc độ đổi mới cao: việc áp dụng Openflow cho phép các nhà khai thác mạng lập trình lại mạng để có thể đạt được các nhu cầu kinh doanh và yêu cầu người dùng cụ thể khi có sự thay đổi.

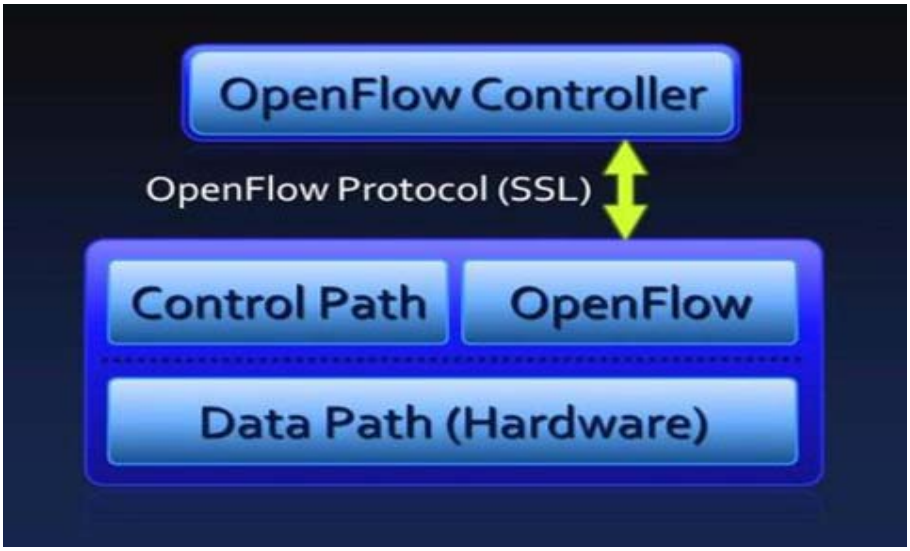
+ Gia tăng độ tin cậy và khả năng an ninh của mạng: các nhà quản trị mạng có thể định nghĩa các trạng thái cấu hình và chính sách ở mức cao, áp dụng tới cơ sở hạ tầng thông qua OpenFlow. Kiến trúc SDN dựa trên cơ sở OpenFlow cung cấp sự điều khiển và tầm nhìn hoàn chỉnh trên mạng nên nó có thể đảm bảo điều khiển truy nhập, định hình lưu lượng, QoS, bảo mật và các chính sách đó được thực thi nhất quán trên toàn bộ cơ sở hạ tầng mạng.

+ Đáp ứng nhu cầu của người dùng một cách linh hoạt: bằng việc tập trung hóa điều khiển mạng và tạo ra trạng thái thông tin có sẵn cho các ứng dụng mức cao hơn, kiến trúc SDN trên cơ sở OpenFlow có thể đáp ứng tốt hơn cho các nhu cầu thay đổi của người dùng.

2.3.Nguyên lý hoạt động

Trong hầu hết các thiết bị chuyển mạch Ethernet hiện đại đều sử dụng các bảng dòng chảy (Flow tables). Những bảng này giúp cho việc chuyển các gói tin từ nơi gửi đến nơi nhận một cách hiệu quả. Mỗi nhà cung cấp sẽ có một bảng dòng chảy riêng, tuy nhiên ta vẫn có thể tách ra một tập hợp các chức năng, được xem là điểm chung cho tất cả các thiết bị chuyển mạch. Ví dụ : QoS hay các báo cáo về lưu lượng. Và OpenFlow thực hiện chuẩn hóa những điểm chung này vào trong một bảng riêng.

Như trên hình 2.1, OpenFlow tách rời các chức năng của lớp truyền dữ liệu và lớp điều khiển ra khỏi nhau. Chức năng liên quan đến truyền dữ liệu vẫn được thực hiện trên thiết bị chuyển mạch như cũ, còn các quyết định về định tuyến cấp cao trong OpenFlow thì do bộ điều khiển (Controller) thực hiện. Controller và các thiết bị chuyển mạch giao tiếp với nhau thông qua giao thức OpenFlow Switching Protocol.



Hình 2.1: Sơ đồ quan hệ giữa Controller và thiết bị Openflow switch

Controller có thể ra lệnh cho các switch thực hiện các quy tắc dành cho các luồng dữ liệu mạng. Những quy tắc này có thể là : truyền dữ liệu theo tuyến đường nhanh nhất, hoặc theo tuyến đường có ít hops nhất…

OpenFlow cung cấp giao diện API duy nhất, nhờ giao diện này người quản trị có thể lập trình công việc của mạng, và đồng thời có thể thiết lập các quy tắc định tuyến gói tin, cân bằng tải, điều khiển truy nhập… Giao diện API này bao gồm 2 thành phần chính: Giao diện lập trình dành cho việc kiểm soát chuyển tiếp gói tin qua các bộ chuyển mạch mạng và bộ các giao diện toàn cầu (global interface), trên cơ sở những giao diện này có thể tạo ra các công cụ quản lý cấp cao.

2.4. Ưu điểm của Openflow

- OpenFlow có một loạt các ưu điểm quan trọng như sau:

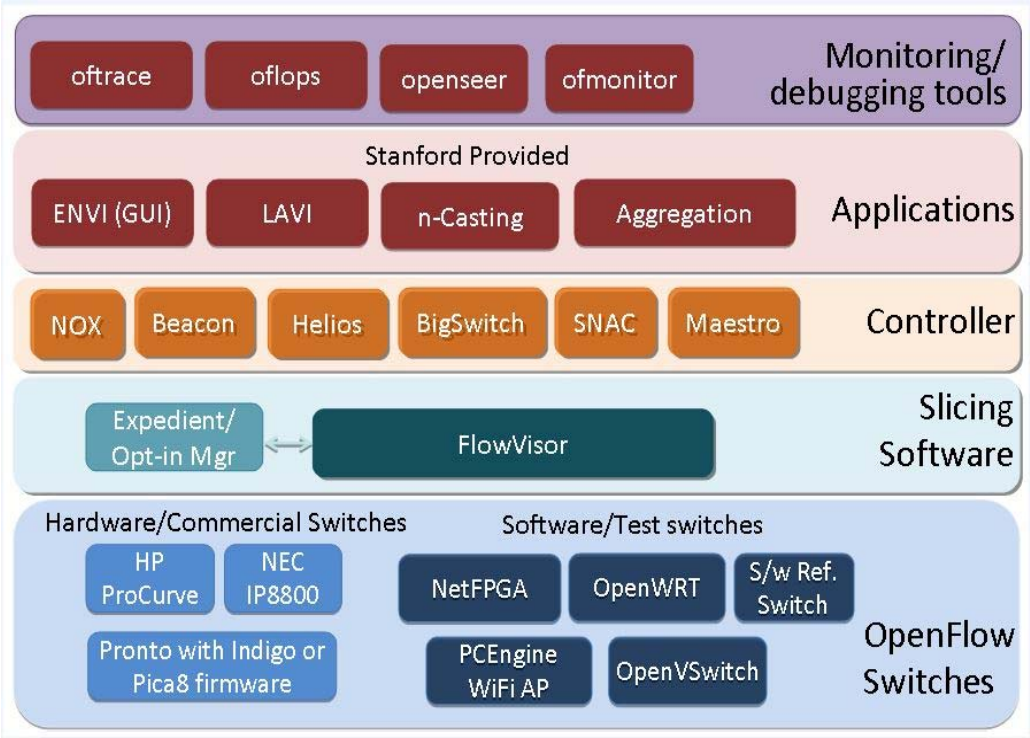
+ Hiệu suất và chi phí: Nhờ việc tách quá trình điều khiển và xử lý ra khỏi thiết bị chuyển mạch, OpenFlow cho phép những thiết bị này tận dụng toàn bộ tài nguyên của mình cho việc tăng tốc chuyển tiếp gói tin. Đồng thời nhờ ảo hóa sự điều khiển mạng, OpenFlow làm giảm chi phí trong việc xây dựng và hỗ trợ mạng

+ Thực hiện và thử nghiệm các chức năng mới: Công cụ phần mềm của OpenFlow cho phép người quản trị thêm chức năng mới vào kiến trúc mạng hiện có. Nhờ đó các chức năng mới sẽ làm việc trên nhiều nền tảng mà không cần tái thực hiện trong các firmware của thiết bị chuyển mạch của mỗi nhà cung cấp. Nhờ giao diện mở API, công nghệ OpenFlow cũng cho phép người quản trị viên hay lập trình viên tạo ra các phần mềm quản lý bất kỳ, từ đó thử nghiệm chức năng mới của thiết bị chuyển mạch. Trước đây, công việc này là rất khó khăn, vì các bộ định tuyến hay chuyển mạch được sản xuất bởi các nhà cung cấp khác nhau không có một giao diện API chung.

+ Bảo mật và quản lý dễ dàng: Trên bộ điều khiển trung tâm (Controller) của OpenFlow, người quản trị có thể quan sát toàn bộ mạng dưới một cái nhìn duy nhất, nhờ đó tăng sự đơn giản trong điều khiển, hỗ trợ bảo mật và thực hiện các nhiệm vụ khác. Vì OpenFlow cho phép quản trị viên thấy rõ tất cả các luồng dữ liệu nên họ sẽ dễ dàng phát hiện sự xâm nhập trái phép hay làm rõ các vấn đề khác. OpenFlow đồng thời cũng cho phép người quản trị hệ thống thiết lập các ưu tiên đối với những dạng luồng dữ liệu khác nhau và phát triển các chính sách phù hợp cho mạng khi có sự cố tắc nghẽn hay các vấn đề khác với thiết bị. Ngoài ra, công nghệ OpenFlow hứa hẹn khả năng tạo ra cấu trúc mạng ảo, xây dựng theo yêu cầu các mạng LAN và WAN ảo mà ko cần thay đổi cấu trúc phần cứng của mạng. Để thực hiện điều này cần phải xem xét khả năng tạo ra mặt phẳng điều khiển trung tâm ảo, hỗ trợ các chức năng quản lý mạng. Chức năng này có thể đặc biệt có ích cho việc điều khiển trung tâm xử lý dữ liệu. Ví dụ nhờ controller OpenFlow, người quản trị mạng có thể tạo ra mạng LAN ảo cho một khách hàng mới mà không cần phải thay đổi trong từng bộ chuyển mạch hay nhóm thiết bị chuyển mạch của một nhà cung cấp nhất định.

+ Điện toán đám mây: Theo những người ủng hộ OpenFlow thì công nghệ này có khả năng hỗ trợ tốt các mức độ “thông minh” mong muốn của mạng cho điện toán đám mây.

Trên hình 2.2 mô tả bộ các phần mềm và phần cứng hỗ trợ hoặc có thể tham gia vào việc xây dựng mạng SDN với việc sử dụng OpenFlow. Các nhiệm vụ chính được giao cho các thiết bị chuyển mạch và OpenFlow controller. Để giúp đỡ các nhà phát triển hay nghiên cứu, người ta đã tạo ra các ứng dụng viết sẵn, những ứng dụng này sẽ thực hiện các chức năng của mạng và các công cụ để gỡ lỗi, giám sát.



Hình 2.2: Bộ các phần mềm và phần cứng hỗ trợ SDN và OpenFlow

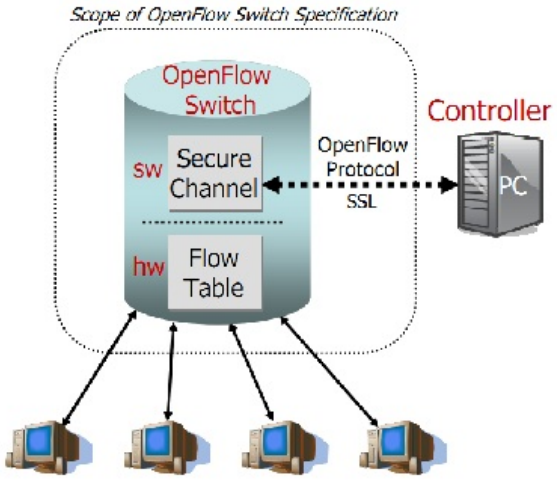
2.5. Các khái niệm và thành phần cơ bản

- Các thành phần cơ bản của mạng SDN trên cơ sở giao thức OpenFlow là:

+ OpenFlow switch.

+ Controller.

Trong đó controller giao tiếp với các OpenFlow Switch thông qua giao thức OpenFlow.

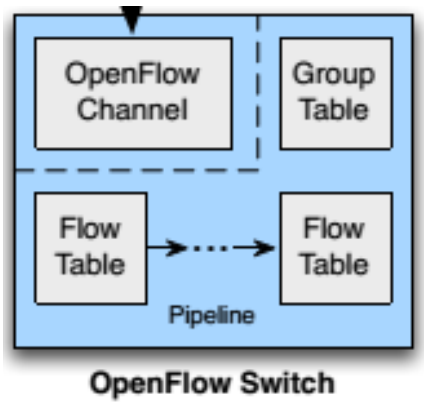


Hình 2.3: Sơ đồ tương tác giữa switch và controller theo giao thức OpenFlow

2.5.1. OpenFlow Switch

OpenFlow Switch bao gồm Group table, các Flow table và một kênh OpenFlow channel.

Trong đó các Flow table và Group table chứa các thông tin do Controller cung cấp để thực hiện định tuyến các gói tin. OpenFlow channel là kênh đảm nhiệm nhiệm vụ liên lạc giữa controller và openflow switch.



Hình 2.4: Cấu trúc OpenFlow Switch

2.5.1.1.Các khái niệm cơ bản

- Trong phần này chúng ta sẽ xem xét các khái niệm, thuật ngữ cơ bản được sử dụng để mô tả nguyên lý hoạt động của giao thức OpenFlow và các thành phần chính của nó.

+ Packet: Là gói tin, bao gồm tiêu đề và các dữ liệu có ích (payload).

+ Pipeline: Tập hợp các bảng liên kết với nhau. Chúng hỗ trợ việc kiểm tra tiêu đề của gói tin, chuyển tiếp gói tin hay chỉnh sửa gói tin trong bộ chuyển mạch OpenFlow.

+ Port: Là nơi mà các gói tin được gửi đến hoặc gửi đi trong OpenFlow switch. Port có thể là vật lý hoặc cổng ảo được xác định bởi switch hay cổng ảo được xác định bởi giao thức OpenFlow.

+ Match Field: Là trường thông tin được xét trong packet và bao gồm tiêu đề của packet, cổng vào, giá trị của metadata.

+ Metadata: Là kiểu dữ liệu chứa thông tin về các dữ liệu khác, sử dụng để chuyển thông tin từ bảng này đến bảng khác.

+ Instruction: Chứa một loạt các hành động được áp dụng đến các packet hoặc để chỉnh sửa quá trình xử lý packet trong pipeline.

+ Flow Entry: Là một phần tử ở trong Flow table được sử dụng để kiểm tra và xử lý các gói tin. Nó bao gồm một tập hợp các trường như Match, metadata, counter , instruction..

+ Flow table: Một bảng chứa các flow entry.

+ Action: Là hoạt động chuyển tiếp packet đến port hoặc chỉnh sửa packet (ví dụ giảm giá trị của trường TTL). Các hành động này có thể được xác định như một phần tập hợp các Instruction liên quan đến flow hoặc hoạt động của pipeline.

+ Action Set: Tập hợp các hành động liên quan đến packet. Tập hợp hành động này được tích tụ liên tục khi diễn ra quá trình xử lý ở mỗi bảng trong pipeline, và nó được thực hiện khi packet đi ra khỏi pipeline.

+ Group: Danh sách các "container" của các hành động và các phương pháp nào đó để lựa chọn một hoặc nhiều container này cho việc áp dụng chúng vào từng packet của nhóm.

2.5.1.2. Flow table

Một flow table bao gồm các thành phần sau:

- Trong đó:

+ Match Fields: Dùng để so khớp với các gói tin. Nó bao gồm các cổng vào, tiêu đề gói tin và tùy chọn dữ liệu được quy định theo bảng trước đó.

+ Trong bảng 2.1 chỉ ra các trường cơ bản và các trường này dùng để so sánh với các packet đi vào. Mỗi trường sẽ chứa một giá trị nhất định, hoặc một giá trị ANY. Giá trị ANY này sẽ trùng khớp với bất kỳ giá trị nào. Ngoài việc sử dụng các tiêu đề packet để so sánh, thì có thể sử dụng thêm port vào và trường metadata. Metadata dùng để truyền thông tin giữa các bảng trong thiết bị chuyển mạch.

Bảng 2. 1: Các entry thuộc trường match.

Trường	Số Bit
Ingress Port	32
Metadata	64
Ethernet source address	48
Ethernet destination address	48
Ethernet type	16

VLAN ID	12
VLAN priority	3
MPLS label	20
MPLS traffic class	3
Ipv4 source address	32
Ipv6 destination address	32
Ipv4 protocol/ ARP opcode	8
Ipv4 ToS bit	6
Transport source port/ ICMP type	16
Transport destination port/ ISMP code	16

+ Priority: Trường để so sánh sự ưu tiên của flow entry.

+ Counters: Trường này sẽ được cập nhật khi gói tin được so khớp.

+ Instructions: Trường chỉ các lệnh tương ứng với bản tin, dùng để chỉnh sửa các hành động hoặc quá trình xử lý pipeline.

+ Timeouts: Thời gian chờ trước khi gói tin bị hết hạn.

+ Cookie: Là phần dữ liệu được lựa chọn bởi bộ điều khiển. Bộ điều khiển có thể sử dụng nó để lọc thống kê lưu lượng, thay đổi lưu lượng và xóa lưu lượng.

2.5.1.3. Group Table

- Group table chứa các thành phần như sau:

Bảng 2. 2: Các thành phần cơ bản của entry trong Group Table

Group Identifier	Group Type	Counters	Action Buckets

- Mỗi mục của Group Table chứa định danh của nhóm, loại của nhóm, counters và danh sách các hoạt động pipeline.

+ Định danh nhóm (Group Identifier) là một số nguyên không dấu 32 bit, dùng để xác định nhóm.

+ Loại của nhóm (Group Type): xác định loại của nhóm.

+ Counters: Giá trị của trường này được cập nhật mỗi khi có packet được xử lý bởi nhóm.

+ Danh sách các hoạt động pipeline: Danh sách thứ tự các hoạt động của pipeline, trong đó mỗi hoạt động pipeline chứa một bộ các hành động dùng để thực hiện hay chỉnh sửa các tham số liên quan đến chúng.

- Có các loại nhóm xác định sau:

+ All: Tất cả các hoạt động pipeline trong nhóm đều được thực hiện. Nhóm này được sử dụng cho các gói tin multicast và broadcast.

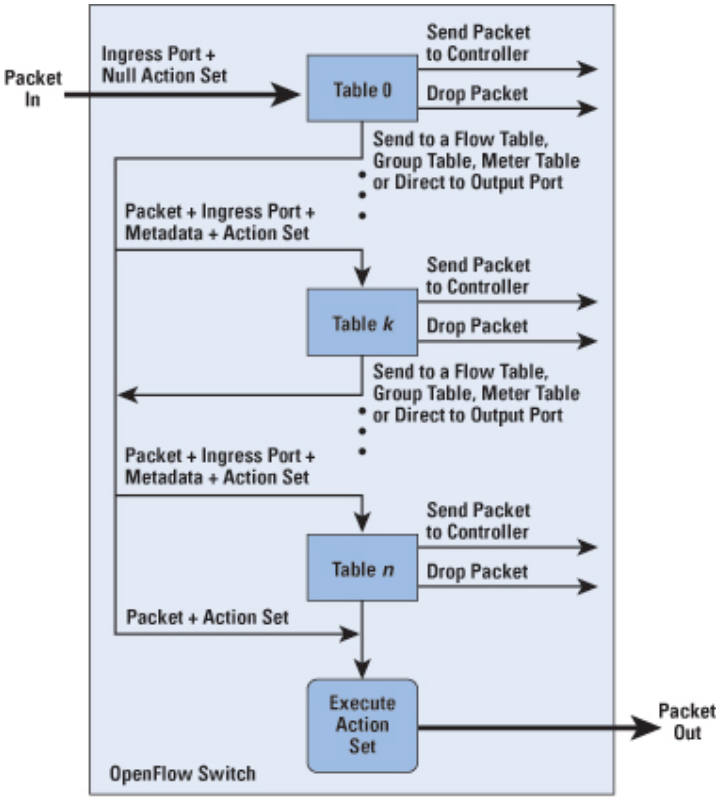
+ Select: Chỉ một hoạt động pipeline trong nhóm được thực hiện. Các packet được gửi vào một pipeline trong nhóm dựa trên một thuật toán lựa chọn, được tính toán bởi bộ chuyển mạch.

+ Indirect: Chỉ một pipeline hoạt động đã xác định từ trước được thực hiện trong bảng.

+ Fast failover: Hoạt động pipeline đầu tiên sẽ được thực hiện. Nếu không có các pipeline “còn sống” thì các packet sẽ bị loại bỏ.

2.5.1.4. Quá trình xử lý pipeline

Pipeline của mỗi bộ chuyển mạch OpenFlow chứa tập hợp các bảng dòng chảy Flow Table. Mỗi bảng này lại chứa tập hợp các entry. Quá trình xử lý các gói tin (packet) bằng pipeline chính là xác định cách mà các gói tin tương tác với những bảng dòng chảy này (Hình 2.5). Nếu bộ chuyển mạch OpenFlow chỉ có 1 flow table thì trong trường hợp này pipeline processing sẽ được giản lược đi rất nhiều.



Hình 2.5: Quá trình xử lý pipeline

Các Flow table của bộ chuyển mạch OpenFlow được đánh số thứ tự bắt đầu từ 0. Quá trình xử lý pipeline sẽ bắt đầu từ bảng 0. Các Flow table khác có thể được sử dụng tùy thuộc vào kết quả của việc so sánh các trường trong gói tin với các entry trong bảng đó.

Nếu gói tin tương ứng với entry trong flow table, thì bộ các instruction tương ứng sẽ được thực hiện. Các instruction trong entry sẽ hướng gói tin đến một bảng khác (sử dụng chỉ dẫn: Goto), ở đó quá trình xử lý packet tương tự sẽ được diễn ra. Lưu ý là entry chỉ có thể chuyển packet đến một bảng flow table khác với số thứ tự lớn hơn số thứ tự của bảng hiện tại. Entry của bảng cuối cùng không thể chứa chỉ dẫn Goto. Nếu khi xét các entry mà không diễn ra việc chuyển gói tin sang một bảng khác thì quá trình xử lý pipeline sẽ kết thúc tại đó. Và khi quá trình xử lý pipeline dừng lại, gói tin được xử lý tương ứng với các bộ hành động liên quan đến nó, thông thường sẽ là chuyển tiếp tiếp tục đến một thiết bị chuyển mạch khác.

Nếu gói tin khi được xét nhưng không tìm thấy entry tương ứng trong bảng, thì người ta gọi đó là trường hợp “miss – trượt”. Trong trường hợp này các hành động tiếp theo sẽ tùy thuộc vào cấu hình của bảng. Theo mặc định thì packet sẽ được chuyển đến controller theo kênh điều khiển nhờ bản tin dạng packet_in, một phương án khác là loại bỏ gói tin đó đi. Flow Table cũng có thể chỉ ra rằng nếu “miss” diễn ra thì việc xử lý gói tin vẫn phải tiếp tục, lúc này gói tin có thể được chuyển đến bảng có số thứ tự tiếp theo để xử lý.

2.5.1.5. Một số hoạt động trong OpenFlow switch

- Thiết bị chuyển mạch là phần quan trọng của mạng. Trong thành phần của bộ chuyển mạch gồm các thành phần như sau:

+ Flow Tables.

+ Group Tables.

Các câu lệnh điều khiển chính của controller được gửi đến bộ chuyển mạch bao gồm:

+ Thêm flow.

+ Cập nhật flow.

+ Xóa flow.

- Các trạng thái làm việc chính:

+ Bị động (Sẽ đáp ứng lại từng gói tin của mạng mà không chuẩn bị hay cấu hình trước)

+ Chủ động (Được chuẩn bị hay cấu hình trước, khi gói tin đến thì thực hiện theo những gì đã cài đặt sẵn).

- Mỗi bảng Flow Table trên bộ chuyển mạch chứa một hoặc nhiều entry tương ứng với các dòng chảy dữ liệu nhất định:

+ Match field.

+ Counters.

+ Instructions.

- Việc kiểm tra và so sánh gói tin với các bảng được bắt đầu từ bảng đầu tiên và có thể tiếp tục trong các bảng tiếp theo. Nếu phát hiện có sự trùng khớp thì sẽ thực hiện các chỉ dẫn đã có sẵn trong bảng tương ứng với từng entry. Nếu không tìm thấy entry tương ứng thì bộ chuyển mạch thực hiện một trong các việc sau:

+ Gửi packet đến controller để controller phân tích.

+ Loại bỏ packet.

+ Tiếp tục tìm kiếm trong bảng tiếp theo.

- Bộ các chỉ dẫn chứa trong mỗi entry sẽ mô tả các hành động mà bộ chuyển mạch phải làm trong trường hợp có sự trùng khớp khi so sánh. Bộ chỉ dẫn đó bao gồm:

+ Chuyển tiếp gói tin.

+ Chỉnh sửa tiêu đề gói tin.

+ Gửi gói tin đến Group Table để xử lý.

+ Gửi gói tin đến pipeline processing để xử lý.

- Việc chuyển tiếp gói tin có thể là gửi gói tin đến:

+ Port vật lý của bộ chuyển mạch.

+ Port ảo của bộ chuyển mạch.

+ Port dành riêng của bộ chuyển mạch.

- Port ảo của bộ chuyển mạch có thể được sử dụng để xác định chính xác nhóm của các kênh tổng hợp, đường hầm, đồng thời xác định cả các port có thông tin phản hồi (loopback).

- Port dành riêng của bộ chuyển mạch được xác định bởi giao thức OpenFlow. Các port dành riêng ảo này có thể được sử dụng để mô tả các quy tắc chung cho việc chuyển tiếp packet.

- Xử lý trong Group Table được sử dụng để thực hiện các hành động phụ đối với gói tin. Chính các nhóm sẽ chứa các bộ hoạt động với các yêu cầu phức tạp hơn (ví dụ : Đa đường, định tuyến nhanh,…).

- Pipeline processing cho phép chuyển tiếp gói tin vào bảng tiếp theo hoặc truyền các thông tin dịch vụ giữa các bảng dưới dạng metadata.

- Những nhà sản xuất các bộ chuyển mạch có thể tự mình lựa chọn phương pháp thực hiện cấu trúc bên trong của thiết bị, tuy nhiên quá trình xem xét và so sánh gói tin cũng như là các quy tắc xử lý gói tin cần phải giống nhau.

2.5.2. Controller

Controller trong khái niệm OpenFlow là yếu tố cơ bản và là trung tâm của mạng SDN, trong đó tập trung tất cả các chức năng điều khiển mạng SDN. Hệ điều hành trên controller chính nó không điều khiển mạng, mà nó chỉ giúp cho giao diện lập trình API điều khiển mạng. Vì thế về cơ bản, việc giải quyết các bài toán điều khiển mạng được thực hiện nhờ các ứng dụng được triển khai trên cơ sở API của hệ điều hành controller. Cần lưu ý rằng giao diện lập trình này phải đủ tính tổng quát để hỗ trợ được nhiều ứng dụng từ đó có thể giải quyết được các vấn đề điều khiển mạng.

Khác với cách giải thích thông thường thuật ngữ NOS (Network Operation System – Hệ điều hành mạng) là một hệ điều hành tích hợp với các bộ giao thức mạng, trong trường hợp này thuật ngữ NOS được hiểu là một hệ thống phần mềm hỗ trợ giám sát, truy nhập, điều khiển các tài nguyên của toàn bộ mạng chứ không

phải là của từng thiết bị. NOS sẽ hình thành các dữ liệu về trạng thái của tất cả tài nguyên mạng và hỗ trợ các ứng dụng điều khiển truy nhập vào chúng.

- Giao diện API của hệ điều hành mạng có các đặc tính cơ bản sau:

+ Thứ nhất, giao diện API đó cung cấp khả năng tạo ra các ứng dụng trên cơ sở mô hình điều khiển tập trung, nghĩa là các ứng dụng được viết ra sao cho toàn bộ mạng được biểu diễn trên một bộ máy.

+ Thứ hai, giao diện API cung cấp khả năng tạo ra các ứng dụng ở mức ảo hóa cao (ví dụ tên người sử dụng, tên của host), chứ không phải là các tham số cấu hình cấp thấp (ví dụ IP, MAC address). Điều này cho phép thực hiện các câu lệnh điều khiển mà không phụ thuộc vào topo cơ bản của mạng. Dĩ nhiên việc làm này yêu cầu một sự ánh xạ tương ứng giữa lớp ảo hóa cấp cao và các cấu hình cấp thấp.

Các thiết bị chuyển mạch OpenFlow là đối tượng chịu điều khiển gián tiếp từ NOS. Controller về nguyên tắc sẽ làm việc trên một server kết nối tới mạng, và có thể là: một controller điều khiển toàn bộ các switches OpenFlow; hoặc một controller điều khiển một bộ các switch cụ thể nào đó; hoặc có thể là một controller điều khiển một switch trong mạng. Controller hỗ trợ giao diện để tạo ra, chỉnh sửa, xóa bỏ, điều khiển các cấu hình trong các bảng flow table của các switches OpenFlow.

Có một điều rất quan trọng khi sử dụng nhiều controller trong mạng SDN đó là tất cả các controller phải có chung một topo mạng trong mọi thời điểm. Topo mạng đó có thể là topo về các switche, hoặc sự phân bố người dùng, host hay các thành phần, dịch vụ khác của mạng. Vì thế một trong những bài toán quan trọng nhất được xử lý bởi NOS là giám sát mạng thường xuyên để xây dựng topo mạng một cách thống nhất.

Ở thời điểm hiện tại, người ta đã tạo ra nhiều controller cho SDN, ví dụ: NOX, Beacon, Maestro, Trema, SNAC, Helios, BigSwitch. Tuy nhiện việc cùng lúc sử dùng nhiều controllers khác loại với nhau vẫn chưa thể thực hiện được.

2.5.3. OpenFlow protocol

- Giao thức OpenFlow mô tả quá trình trao đổi thông tin giữa OpenFlow Controller và OpenFlow Switch. Giao thức OpenFlow cho phép bộ điều khiển thực hiện các thao tác như thêm, cập nhập, chỉnh sửa và xóa các flow entry ở trong flow table. Nó hỗ trợ 3 loại bản tin như sau:

+ Controller đến Switch: bản tin này được bắt đầu bởi controller, trong mô số trường hợp thì nó được bắt đầu bởi switch. Bản tin này cho phép controller quản lý trạng thái của switch bao gồm các cấu hình và chi tiết các luồng tin và các entry của flow table. Cũng như gói tin thông điệp đầu ra, bản tin này được sử dụng khi switch gửi các bản tin đến controller và controller quyết định không hủy bản tin mà đưa nó ra port đầu ra của switch.

+ Bản tin không đồng bộ(Asynchronous): loại bản tin này được gửi mà không cần sự đồng ý từ bộ điều khiển. Loại này bao gồm các thông báo khác nhau đến bộ điều khiển. Ngoài ra còn có các bản tin packet-in, bản tin này được sử dụng bởi switch để gửi gói tin đến controller khi bản tin không khớp với trường nào ở flow table.

+ Bản tin đối xứng (Symmetric): bản tin này được gửi đi mà không cần sự đồng ý của controller hoặc switch. Nó tuy đơn giản nhưng rất hữu ích.Ví dụ như bản tin hello thường được gửi qua lại giữa controller và switch khi trạng thái kết nối lần đầu tiên được thiết lập. Bản tin echo và bản tin phúc đáp có thể được sử dụng bởi switch hoặc controller để đo độ trễ và băng thông của kết nối giữa controller và switch hoặc để xác minh rằng thiết bị đang hoạt động.

Bảng 2. 3: Các loại bản tin trao đổi giữa controller và switch

Bản tin	**Mô tả**
Controller đến Switch	
Features	Bản tin yêu cầu khả năng của switch.
Configuration	Bản tin cấu hình. Switch trả lời với các

	thông số thiết lập.
Modify-State	Bản tin yêu cầu thêm, xóa, chỉnh sửa các entry và thiết lập các thuộc tính cổng của switch.
Read-State	Bản tin thu thập các thông tin từ các switch ví dụ như các thông tin cấu hình,các khả năng của switch…
Packet-out	Bản tin chỉ gói ra tại một cổng được chỉ định trên switch.
Barrier	Bản tin yêu cầu hoặc trả lời các tin nhắn được sử dụng bởi controller để đảm bảo sự phụ thuộc của tin nhắn đã nhận được cho các hoành động đã hoàn thành.
Role-Request	Bản tin thiết lập hoặc truy vấn vai trò của OpenFlow channel. Hữu dụng khi switch kết nối với nhiều controller.
Asynchronous-Configuration	Bản tin thiết lập các bộ lọc trên các thông điệp khong đồng bộ hoặc truy vấn bộ lọc.
Bản tin không đồng bộ	
Packet-in	Đưa gói tin đến controller.
Flow-Removed	Thông báo cho bộ điều khiển về việc xóa một flow entry trên một flow table.
Port-Status	Thông báo cho controller về sự thay đổi trên một cổng.
Error	Thông báo có lỗi kết nối.
Bản tin đối xứng	

Hello	Bản tin trao đổi giữa controller và switch khi kết nối mới được thiết lập.
Echo	Bản tin được gửi bởi controller hoặc switch và khi nhận được thì bên nhận phải đáp ứng trở lại.
Experimenter	Bản tin dùng cho chức năng bổ sung.

2.6. Kết luận chương

SDN và OpenFlow không phải là một. SDN liên quan đến khái niệm chung về cấu trúc mạng, trong đó tách rời phần điều khiển và phần chuyển tiếp các gói tin. Còn OpenFlow là một giao thức, nó được tạo ra, chuẩn hóa, để hiện thực hóa khái niệm SDN. Nó cho phép chúng ta dễ dàng triển khai các giao thức chuyển mạch và định tuyến trong mạng, cung cấp cho chúng ta một phương thức quản lý mạng hiệu quả và bảo mật cao.

Trong chương này chúng ta đã có cái nhìn tổng quát về giao thức OpenFlow, các thuật ngữ cơ bản, cấu trúc, chức năng của các thiết bị định tuyến OpenFlow, cũng như nguyên lý hoạt động, cách thức xử lý gói tin của OpenFlow Switch. Với các giao diện mở API, giao thức OpenFlow cho phép người lập trình có thể tạo ra các công cụ quản lý mới, ứng dụng mới phù hợp nhu cầu của người sử dụng. OpenFlow cho phép thực hiện các kết nối tới các thiết bị của các hãng khác nhau mà không gây một sự khó khăn nào.

Giao thức OpenFlow hứa hẹn là một giao thức hỗ trợ tốt cho kiến trúc SDN, tuy nhiên giao thức này vẫn chưa được hoàn thiện và cần phát triển hơn nữa trong tương lai.

CHƯƠNG III: SDN TRONG MẠNG CAMPUS VÀ ỨNG DỤNG VÀO MẠNG NỘI BỘ TRƯỜNG ĐẠI HỌC HÀ NỘI

Mạng nội bộ truyền thống hiện nay là một mạng bao gồm nhiều người sử dụng được kết nối cục bộ trong một hoặc nhiều tòa nhà với nhau như mạng các công ty, doanh nghiệp, bệnh viện … Với sự phát triển nhanh chóng của các hệ thống, sự gia tăng đột biến của việc sử dụng các thiết bị di động và các ứng dụng đang dần ảnh hưởng đến mạng, chúng yêu cầu các hệ thống phải nhanh nhạy hơn, có hiệu suất cao hơn. Các thách thức hiện nay đặt ra cho nó là vô cùng lớn, đòi hỏi sự thay đổi và phát triển để đáp ứng nhu cầu sử dụng hiện nay. Mạng SDN có thể giảm bớt được một phần nào các thách thức này bằng việc cung cấp tính linh hoạt và khả năng phát triển các chức năng một cách nhanh chóng và hiệu quả.

Chương này giới thiệu về một số khó khăn khi sử dụng mạng nội bộ truyền thống hiện nay, việc ứng dụng và triển khai mạng SDN để giải quyết các khó khăn đó. Qua đó mô phỏng một mô hình mạng SDN dựa trên các chương trình giả lập để thấy được sự hiệu quả và lợi ích mà SDN góp phần mang lại.

3.1. Triển khai SDN cho mạng Campus

3.1.1. Giới thiệu

Mạng campus là mạng gồm các người sử dụng được kết nối cục bộ trong một hoặc vài tòa nhà lại với nhau như mạng của các công ty, các trường học, bệnh viện.... Ngày nay nó đang đối mặt với nhiều thách thức lớn. Các thiết bị di động, xu hướng BYOD và số lượng ngày càng tăng của các thiết bị kết nối mạng và các ứng dụng đang dần ảnh hưởng đến mạng, chúng yêu cầu mạng phải nhanh nhạy hơn, hiệu suất cao hơn . Một trong những lý do của những thách thức kể trên là do công nghệ mạng không đuổi kịp với sự phát triển của nhu cầu người sử dụng. Mạng SDN có thể giảm bớt được các thách thức này bằng việc cung cấp tính linh hoạt và khả năng phát triển các chức năng mới một cách nhanh chóng và hiệu quả.

3.1.2. SDN trong mạng campus

3.1.2.1. Các đặc tính và hạn chế của mạng campus

- Ngày nay các mạng campus phải đáp ứng và hỗ trợ sự đa dạng của:

+ Người sử dụng: nhân viên, khách hàng, du khách, sinh viên, giảng viên…

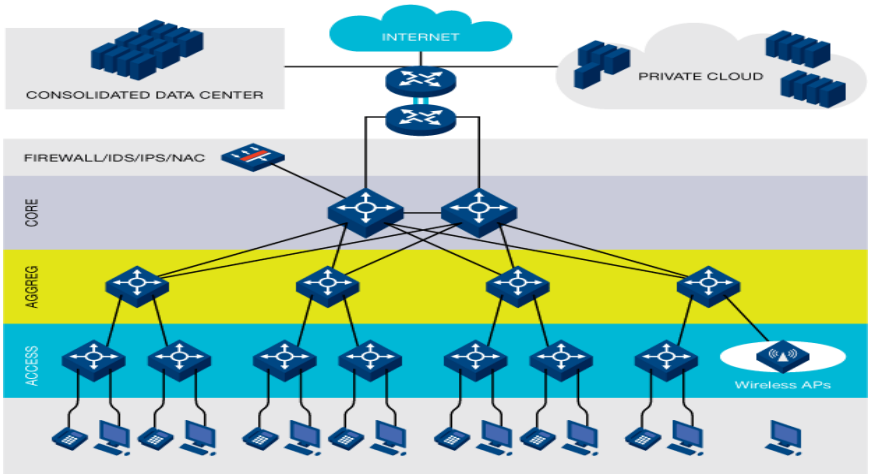
+ Các thiết bị kết nối: điện thoại thông minh, máy tính bảng, máy tính xách tay, máy ảnh, điện thoại IP.

+ Ứng dụng: Các ứng dụng kinh doanh, internet, các ứng dụng chơi game trực tuyến…

+ Phương thức kết nối: Không dây, có dây, 3G/LTE…

Với sự đa dạng và không đồng nhất về các nhu cầu dẫn đến rất khó quản lý hiệu quả. Các quá trình thiết lập cấu hình, triển khai các ứng dụng mới rất mất thời gian và kém linh động.

Kiến trúc cơ bản của một mạng campus học có cấu trúc gồm 3 lớp : Lớp lõi, lớp phân phối và lớp truy cập (như hình dưới). Kiến trúc 3 lớp truyền thống này còn gây ra nhiều khó khăn trong hoạt động như: tránh các vòng loop, hạn chế về kết nối đa đường.



Hình 3.1: Kiến trúc cơ bản của mạng campus

Ngoài ra mạng campus còn cung cấp giải pháp truy cập mạng thông qua các điểm truy cập không dây (Access Point – AP) . Với giải pháp này, lưu lượng truy cập từ các AP được tổng hợp thông qua đường hầm bộ điều khiển và cung cấp của điểm truy cập không dây (Control And Provisioning of Wireless Access Points - CAPWAP) đến các bộ điều khiển truy cập (Access Controller) . Điều này làm cho các bộ điều khiển truy cập trở thành các nút cổ chai trên mạng không dây. Trong khi công nghệ WLAN đã phát triển từ các phiên bản 802.11a/b, 802.11n và 802.11ac làm cho băng thông tăng từ megabit/s lên đến gigabit/s thì vấn đề nút cổ chai AC lại trở nên nghiêm trọng hơn.

- Để giải quyết được các hạn chế trên mạng campus phải có khả năng:

+ Cung cấp các ứng dụng mới phù hợp với các nhu cầu ngày càng tăng của người sử dụng.

+ Quá trình triển khai nâng cấp các ứng dụng mới và quá trình thay đổi cấu hình mạng phải nhanh.

+ Đơn giản hóa và hội tụ cách quản lý mạng không dây và có dây.

+ Cung cấp dịch vụ mới đáp ứng nhu cầu thời gian thực.

Và để đạt được các khả năng đó thì bắt buộc chúng ta phải triển khai một mạng có cấu trúc khác mới các mạng truyền thống, đó là SDN.

3.1.2.2. Triển khai SDN cho mạng campus

- Mạng SDN cho phép chúng ta đơn giản hóa cấu trúc mạng mà vẫn cung cấp được sự linh động cao:

+ Có thể triển khai các dịch vụ một cách nhanh chóng mà không làm ảnh hưởng đến cấu trúc mạng bằng việc áp dụng công nghệ ảo hóa.

+ Cải thiện được chất lượng của các dịch vụ có sẵn nhờ việc tính toán sẵn các con đường định tuyến tối ưu ở một bộ xử lý tập trung.

+ Sử dụng tài nguyên một cách tối ưu vì việc quản lý, dịch vụ và các ứng dụng đều được ảo hóa để sử dụng tối đa trong khi giảm đến tối thiểu không gian và năng lượng tiêu thụ.

Ta xét một ví dụ là một mạng lưới mạng campus gồm nhiều trường đại học phục vụ cho nhiều đối tượng khác nhau như giảng viên, sinh viên, thư viện, cơ sở y tế, sở cảnh sát. Lúc đó mỗi đối tượng sử dụng cần có các chương trình riêng giải quyết các nhu cầu khác nhau và họ phải tuân thủ theo các quy định như PCI, SOX. Điều này đòi hỏi mạng phải cô lập các luồng dữ liệu giữa các người sử dụng khác nhau và tiến hành các mạng logic tức là ảo hóa mạng trên cùng một mạng vật lý.

SDN cho phép các quy tắc được thực thi đối với các ứng dụng khác nhau do đó nó chỉ cho phép truy cập vào tài nguyên mạng cụ thể. Hiện nay để tạo ra các mạng logic trên mạng vật lý người ta sử dụng các công nghệ như MPLS hoặc VRF-Lite. Việc triển khai và sử dụng các công nghệ này rất phức tạp và tốn thời gian. Trong khi đó mạng SDN cho phép tạo ra các lớp mạng logic theo nhu cầu một cách nhanh chóng, chỉ trong vài phút thay vì vài tuần và các thiết bị chuyển mạch có thể thực thi các chính sách linh hoạt để điều khiển và hạn chế giữa các mạng logic với nhau.

- Nói tóm lại, ta sẽ đạt được các lợi ích khi áp dụng mạng SDN như sau:

+ Cô lập các luồng dữ liệu thông qua các chính sách áp dụng cho từng luồng tin, tạo điều kiện cho việc bảo mật các luồng tin và đa sử dụng.

+ Tối ưu hóa băng thông qua việc ảo hóa mạng và kiểm soát tập trung trên cơ sở hạ tầng mạng ảo và vật lý. Điều này cả thiện việc sử dụng các thiết bị cá nhân cũng như toàn mạng.

+ Các hoạt động quản lý trở nên đơn giản hơn.

+ Vì sử dụng các cấu trúc mở nên nó sẽ khuyến khích nhiều nhà cung cấp và các nhà lập trình phát triển để hoàn thiện hơn cở sở mạng và phần mềm hỗ trở mạng.

3.2.3. Nhận xét

Tuy mạng campus phải đối mặt với các yêu cầu đa dạng và nhiều sự thách thức như tích hợp công nghệ, cung cấp và thực thi chính sách bảo mật nhưng với SDN nó giải quyết được tất cả các vấn đề đó. Bộ điều khiển SDN có thể tối ưu hóa các hoạt động của mạng campus bằng việc giảm chi phí quản lý, tăng cường khả năng mở rộng. Mạng lưới campus vốn cung cấp cho nhiều loại đối tượng sử dụng và phải được ảo hóa để đảm bảo thực thi các chính sách riêng biệt cho những người sử dụng khác nhau. Mạng SDN cung cấp quá trình ảo hóa mạng và thực thi các chính sách đơn giản hơn nhiều so với phương pháp thông thường dựa trên giao thức MPLS và VRF-Lite.

3.2. Hiện trạng mạng nội bộ của trường Đại học Hà Nội

3.2.1. Hiện trạng mạng nội bộ của trường Đại học Hà Nội

Hệ thống mạng của trường Đại học Hà Nội hình thành từ năm 1998 (khoảng 50 máy tính cá nhân vào internet, chỉ cần 1 dải IP). Hiện nay đã có hàng nghìn thiết bị truy cập mạng Internet (gồm máy tính server và máy tính cá nhân, camera, Smartphon…) yêu cầu mạng phải có nhiều dải IP và có nhiều vùng với các mức bảo mật khác nhau).

Hiện tại hệ thống mạng của trường được chia theo khu vực: khu làm việc, giảng đường, ký túc xá, các dự án, mỗi mạng tạo thành do sự kết nối thiết bị mạng trong một hoặc vài tòa nhà lại để truy cập vào Internet.

Toàn trường hiện đang có 1100 máy trạm được phân bổ như sau:

- 100 máy sử dụng tại nhà A dùng cho các phòng ban và trung tâm.
- 50 máy sử dụng tại nhà B dùng cho văn phòng La Trobe và các dự án.
- 150 máy sử dụng tại nhà C dùng cho các khoa và bộ môn.
- 150 máy sử dụng tại Thư viện.
- 650 máy dùng cho các phòng LAP tại nhà A1, C, D1.

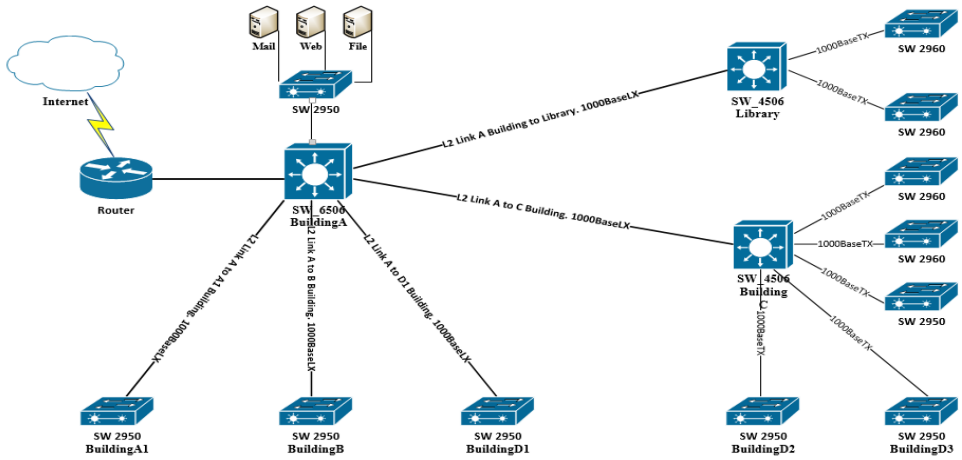
Toàn bộ các máy tính trong trường đều được kết nối ra internet thông qua các máy chủ đặt tại nhà A, nhà C, và Thư viện các máy chủ này chạy hệ điều hành windown 2008 Server, hệ điều hành Linux…. Các máy chủ nội bộ có chức năng chứa dịch vụ của nhà trường: Mail, Web, File, cung cấp DHCP cho các máy trạm theo các Vlan đã định quản lý việc truy cập internet của các máy trạm.

Máy chủ kết nối ra internet thông qua các modem cáp quang tốc độ cao của các nhà cung cấp dịch vụ internet như : FPT, VDC, VNPT… . Khoảng cách từ máy chủ của từng cơ sở tới máy xa nhất là 100m.

Một số Switch được sử dụng tại trường đại học hà nội: Switch Cisco 6506, Switch Cisco 4506, Catalyst 2950, Catalyst 2960.

3.2.2. Mô hình kết nối mạng

Dưới đây là mô hình mạng LAN của trường Đại học Hà Nội.



Hình 3.2: Mô hình mạng LAN của trường Đại học Hà Nội

- Đối với mô hình mạng TCP/IP quy mô lớn như thế này thì ta gặp những vấn đề cơ bản đó là:

+ Thứ nhất là về mặt quản lý mạng. Các thiết bị mạng được phân bố phân tán, người quản trị không thể ngồi một vị trí để có thể quản lý các thao tác cấu hình

hay triển khai một số ứng dụng mới hoặc là mở rộng mạng lưới được. Nếu có vấn đề xảy ra thì không thể biết được quá trình lỗi ở đâu mà phải kiểm tra các cấu hình của từng thiết bị một cách thủ công.

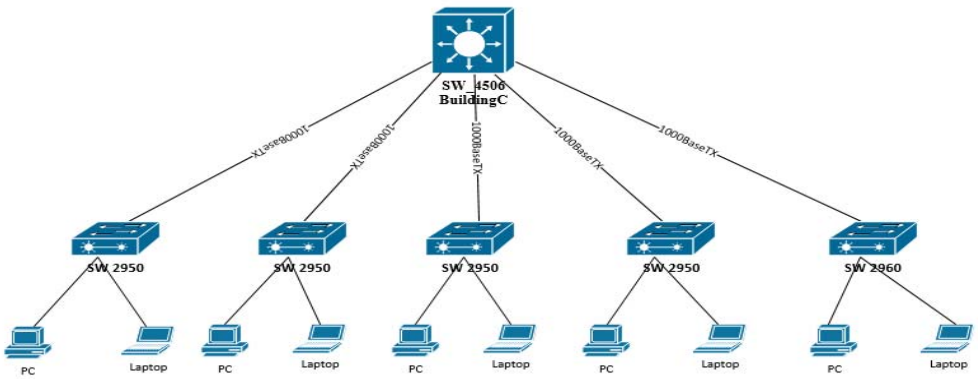
+ Thứ 2 về mặt chi phí thiết bị. Các thiết bị mạng hiện nay thường được tích hợp các phiên bản phần mềm phục vụ các quá trình xử lý gói tin sẵn trong phần cứng, và mỗi thiết bị của từng hãng khác nhau lại có một cơ chế thực hiện riêng. Khi có vấn đề với các thiết bị và không muốn tình trạng bất đồng bộ dẫn đến khó kiểm soát mạng thì người quản trị phải thay thế các thiết bị của cùng một nhà sản xuất bất kể có những thiết bị giá thành rẻ hơn.

- Việc triển khai mạng SDN có thể giúp chúng ta khắc phục được các hạn chế cơ bản nói trên.

Nhưng với mô hình mạng như trên thì việc triển khai SDN gặp rất nhiều khó khăn. Chính vì vậy luận văn chỉ đưa ứng dụng SDN thử nghiệm cho một tòa nhà trong hệ thống mạng trên – mô hình mạng tòa nhà C của trường Đại học Hà Nội, từ đó sẽ cho thấy được những lợi ích mà SDN mang lại.

3.3. Mô hình mô phỏng mạng tòa nhà C, Đại học Hà Nội trên nền tảng SDN

Dưới đây là mô hình mạng LAN của tòa nhà C:



Hình 3.3: Mô hình mạng LAN của tòa nhà C

Tòa nhà C là khu giảng đường, lớp học và văn phòng các khoa, bao gồm 150 máy tính phục vụ cho văn phòng các khoa, 300 máy tính phục vụ cho các lớp học. Toàn bộ các máy tính trong tòa nhà đều được kết nối ra internet thông qua các máy chủ.

3.4. Các công cụ sử dụng trong cấu hình mô phỏng mạng

3.4.1. Mininet

Mininet là một trình giả lập để triển khai mạng lớn trên các nguồn tài nguyên hạn chế của một máy tính đơn giản hay máy ảo, là phần mềm mã nguồn mở miễn phí mà giả lập thiết bị và bộ điều khiển cho phép nghiên cứu trong SDN và OpenFlow. Mininet là một công cụ quan trọng đối với cộng đồng mã nguồn mở SDN bởi nó thường được sử dụng như công cụ để mô phỏng, kiểm tra, xác minh các ứng dụng mới của SDN.

Mininet cho phép tạo topo kích thước quy mô rất lớn, tạo ra một mạng lưới các host, switch, bộ điều khiển và các liên kết ảo, và thực hiện các thử nghiệm trên chúng rất dễ dàng.

Một số chức năng cơ bản của Mininet:

- Cung cấp một mô hình mạng thử nghiệm đơn giản và rẻ (do không tốn kém chi phí mua các OpenFlow Switch) để phát triển các ứng dụng mạng. Do các OpenFlow Switch trong Mininet có tất cả các tính chất mà OpenFlow Switch thật có được nên việc sử dụng mạng ảo hóa bằng Mininet có ý nghĩa về mặt thực tế.

- Cho phép các nhà phát triển ứng dụng làm việc đồng thời, một cách độc lập trên cùng một mô hình mạng.

- Cho phép kiểm thử các mô hình phức tạp mà không cần phải nối dây cho mạng vật lý.

- Cho phép debug và chạy các bài kiểm tra của các mạng lớn, sử dụng CLI.

- Hỗ trợ thiết lập các mô hình tùy biến bất kỳ, gồm tập cơ bản các thông số của mô hình.

- Có thể đem các ứng dụng trên Mininet đi triển khai trên mạng thật với code hoàn toàn không cần thay đổi.

- Cung cấp Python API dễ dàng sử dụng và có khả năng mở rộng.

Ưu điểm của Mininet so với các phương pháp ảo hóa khác như OpenFlowVMS hay Noxrepo.org VM environment gồm có: khởi động nhanh hơn (chỉ tốn vài giây để khởi động một mạng có nhiều OpenFlow switch), tính mở rộng lớn hơn (có thể chứa hàng ngàn nút mạng), cung cấp nhiều băng thông hơn (tổng 2 Gbps với phần cứng bình thường), cài đặt dễ dàng (có sẵn bản VMware để tải về).

Nhược điểm hiện tại của Mininet là chỉ hỗ trợ chạy trên 1 máy tính Linux nên hạn chế về mặt hiệu năng, tuy nhiên trong tương lai gần, nhược điểm này sẽ sớm được khắc phục.

Các thành phần cơ bản trong Mininet:

- Links: các liên kết trong mạng Mininet là một cặp Ethernet ảo hoạt động như một đường dẫn kết nối hai giao diện ảo. Các gói tin được gửi từ giao diện này tới giao diện kia, các giao diện này hoàn toàn giống như các cổng Ethernet cho hệ thống và các phần mềm ứng dụng.

- Hosts: Các Network namespaces chứa trạng thái của mạng (network state). Chúng cung cấp các tiến trình (hoặc các nhóm tiến trình) với quyền điều khiển các giao diện, các cổng và bảng định tuyến. Mỗi host có các giao diện Ethernet riêng (khởi tạo và cài đặt bởi lệnh ip link add/set) và một đường kết nối tới tiến trình Mininet, tiến trình này gửi các lệnh và hiển thị thông tin của mạng.

- Switches: Các switch mềm OpenFlow cung cấp cùng một ngữ nghĩa cho việc gửi các gói tin giống như switch phần cứng.

- Controllers: Controllers có thể ở bất cứ đâu trong mạng thật hoặc trong môi trường mô phỏng.

3.4.2. OpenDayLight

Hiện nay, trên thị trường có khá nhiều bộ điều khiển được sử dụng trong SDN như: OpenDaylight, Floodlight, POX, NOX, ... Trong đó OpenDaylight được sự chú ý rất lớn từ cộng đồng những người quan tâm về công nghệ SDN, vì OpenDaylight có giao diện thân thiện, dễ thao tác, phù hợp cho những người bước đầu tìm hiểu hiểu về SDN.

OpenDaylight là dự án mã nguồn mở hợp tác với Linux Foundation, với mục tiêu là trở thành một thành phần cốt lõi của kiến trúc SDN, cho phép người sử dụng làm giảm sự phức tạp của mạng lưới hoạt động, dễ dàng mở rộng cơ sở hạ tầng mạng. Bộ điều khiển này được viết dựa trên Java, và có hỗ trợ giao thức OpenFlow.

3.4.3. Cài đặt các công cụ mô phỏng

3.4.3.1. Cài đặt mininet

- Bật cửa sổ lệnh Terminal của ubuntu và gõ các lệnh:

git clone git://github.com/mininet/mininet

mininet/until/install.sh –a

Sau khi gõ các lệnh trên thì mininet sẽ được download và cài đặt.

3.4.3.2. Cài đặt Opendayligh controller

- Trên cửa sổ lệnh Terminal ta sử dụng các lệnh sau:

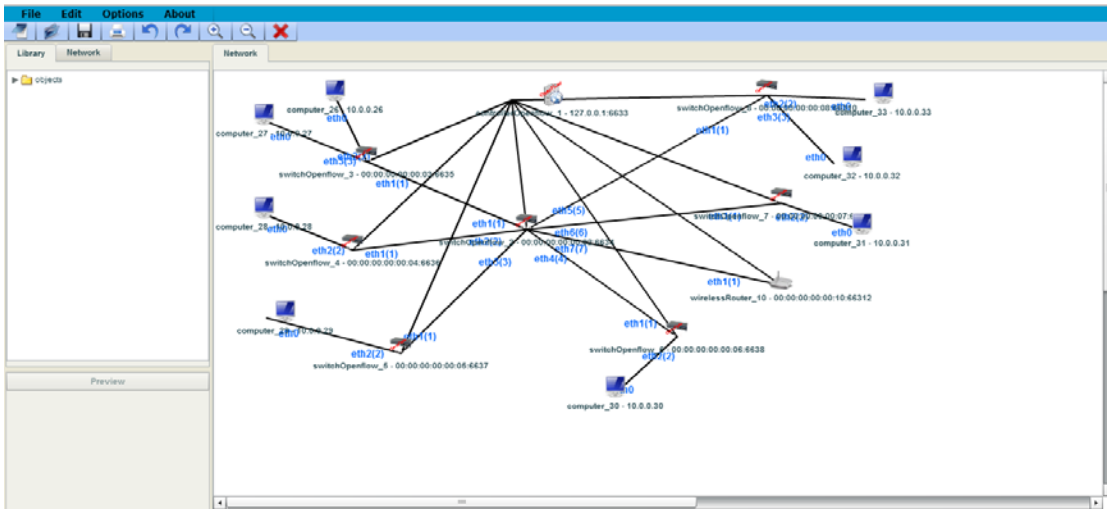
apt-get install maven git openjdk-7-jre openjdk-7-jdk

git clone http://git.opendaylight.org/gerrit/p/controller.git

3.5. Tiến trình thực hiện mô phỏng

3.5.1. Sơ đồ mạng tòa nhà C theo mô hình SDN

+ Hình mô phỏng mạng SDN nhà C Đại học Hà nội.



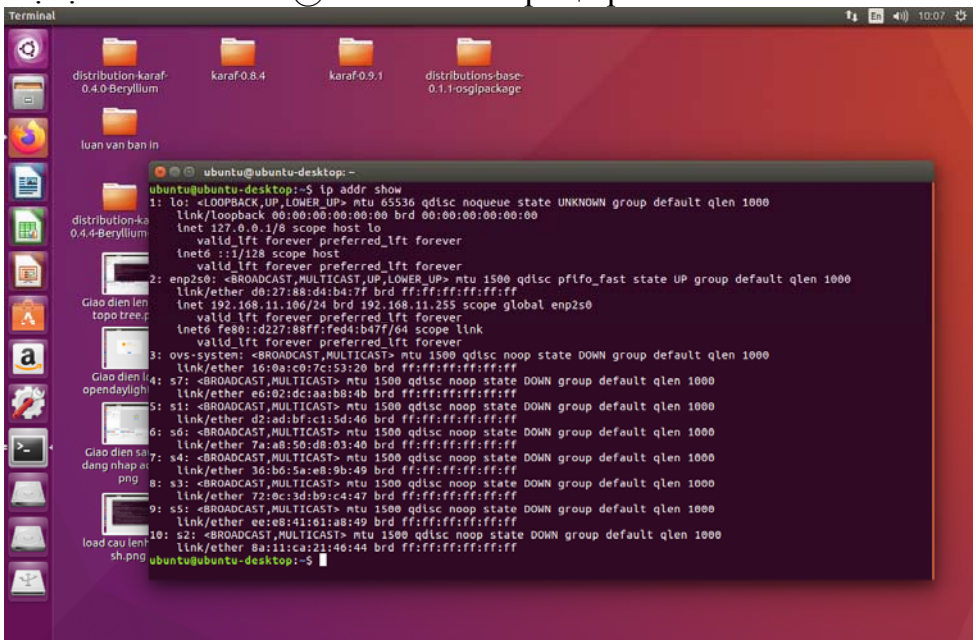
Hình 3.4: Sơ đồ mạng tòa nhà C sau khi vẽ lại theo dạng SDN

3.5.2. Tiến trình mô phỏng

- Khởi động Terminal và trình duyệt web:

+ Sử dụng các câu lệnh sau:

Hiển thị địa chỉ IP: ubuntu@ubuntu-desktop:~$ ip addr show



Hình 3.5: Giao diện lệnh hiện thị IP

Download gói ứng dụng:

ubuntu@ubuntu-desktop:~$ wget

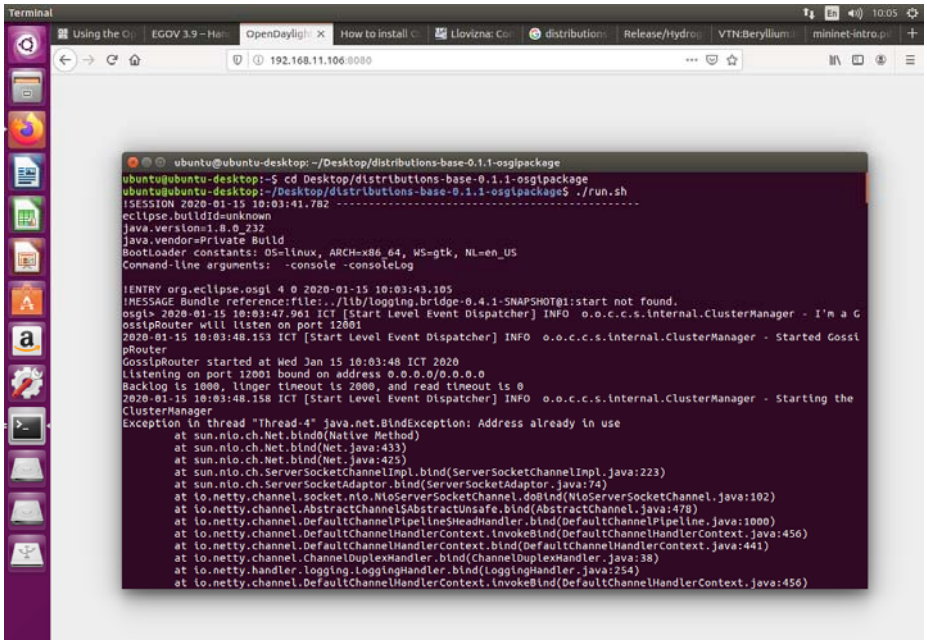
http://nexus.opendaylight.org/content/repositories/opendaylight.release/org/opendaylight/integration/distributions-base/0.1.1/distributions-base-0.1.1-osgipackage.zip

Giải nén file vừa download đến 1 nơi nào đó trên máy tính(ví dụ giải nén đến Desktop). Sử dụng câu lệnh trên terminal để cài đặt và cấu hình Opendaylight:

ubuntu@ubuntu-desktop:~$ cd Desktop

ubuntu@ubuntu-desktop:~$ Desktop cd distributions-base-0.1.1-osgipackage

ubuntu@ubuntu-desktop:~$ Desktop\distributions-base-0.1.1-osgipackage ./run.sh

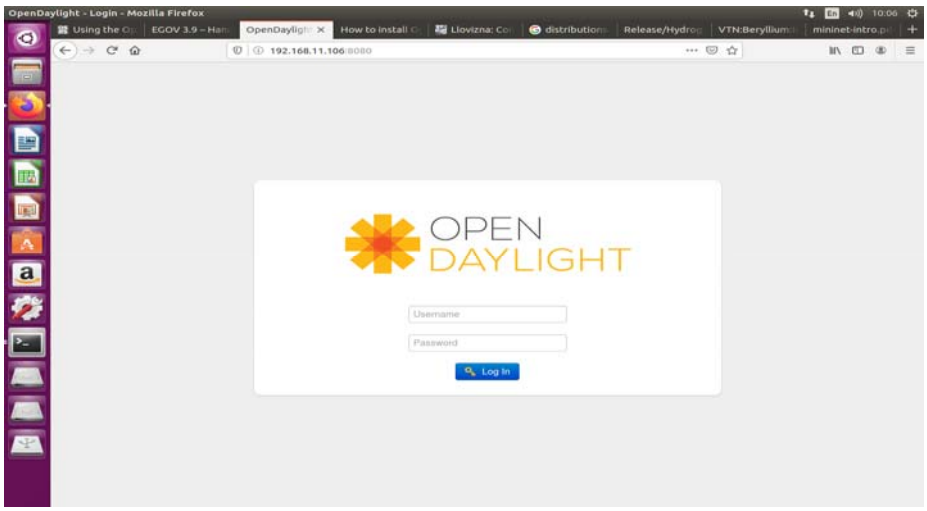


Hình 3.6: Giao diện thực hiện lệnh ./run.sh

./run.sh thực hiện xong (cài đặt và câu hình opendaylight), ta vào trình duyệt web (chrome, ie, firefox) nhập địa chỉ:

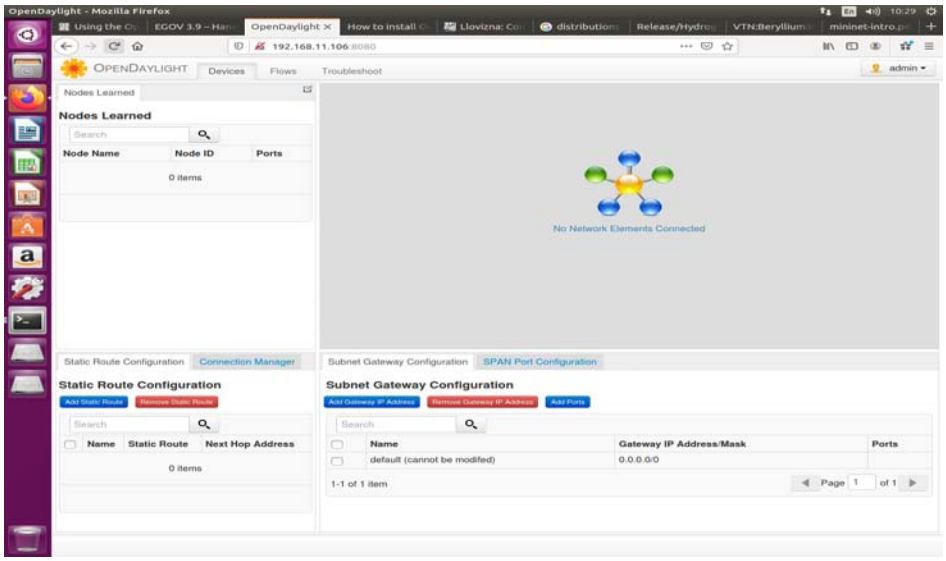
http://<địa chỉ ip>:8080 và nhấn enter

Ví dụ: http://192.168.11.106:8080



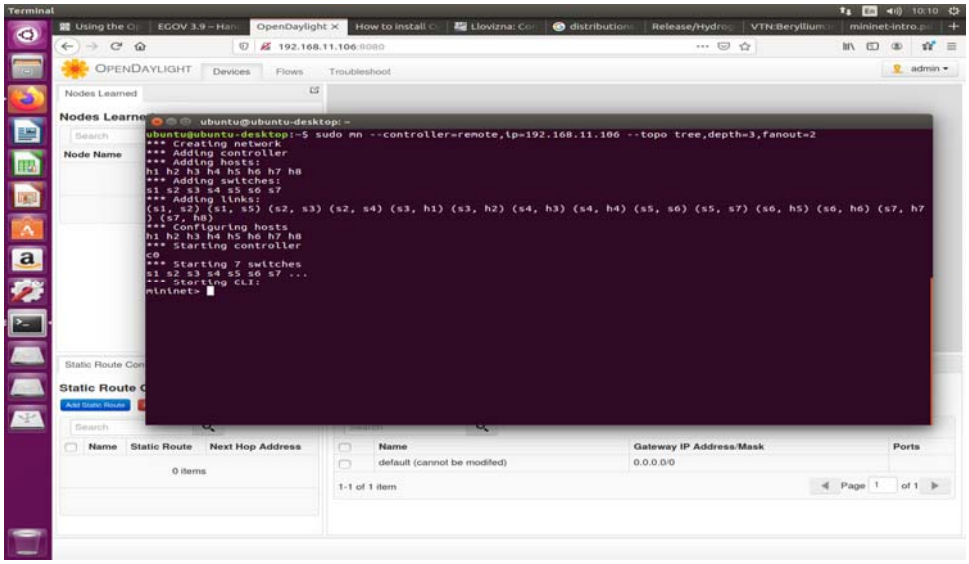
Hình 3.7: Giao diện trang đăng nhập vào Opendaylight Controller

Đăng nhập với account: Username: admin; Password: admin



Hình 3.8: Giao diện trang đăng nhập vào Opendaylight Controller thành công

- Thực hiện câu lệnh tạo topo tree với depth=3, fanout=2



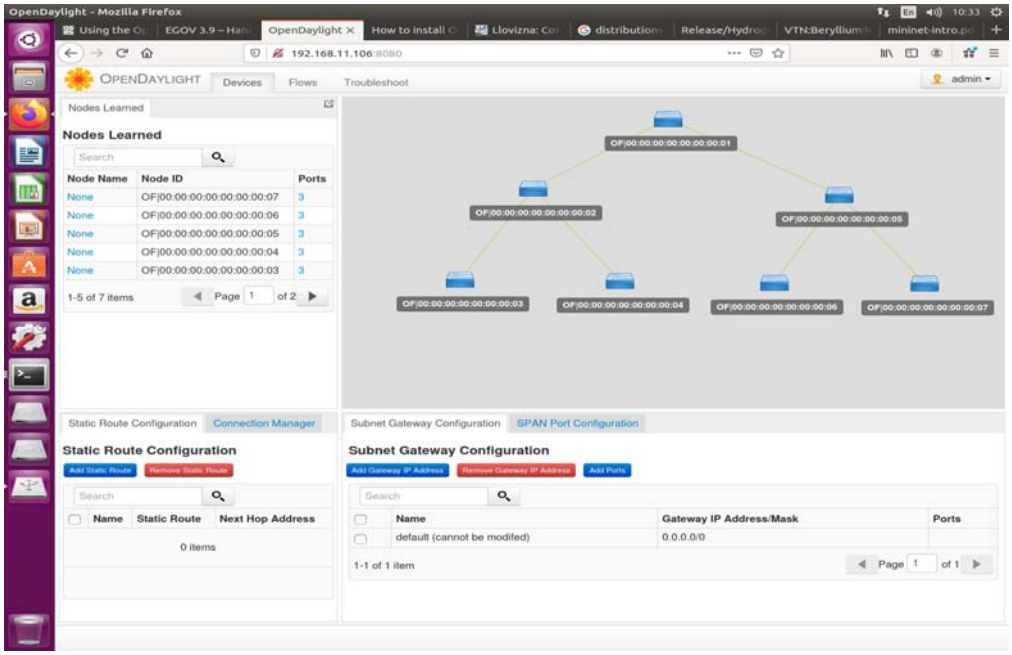
Hình 3.9: Giao diện câu lệnh tạo topo tree

- Hình trên cho ta thấy đã tạo thành công topo tree, sơ đồ gồm các thành phần:

+ Switch: s1, s2, s3, s4, s5, s6, s7.

+ Host: h1, h2, h3, h3, h4, h5, h6, h7,h8.

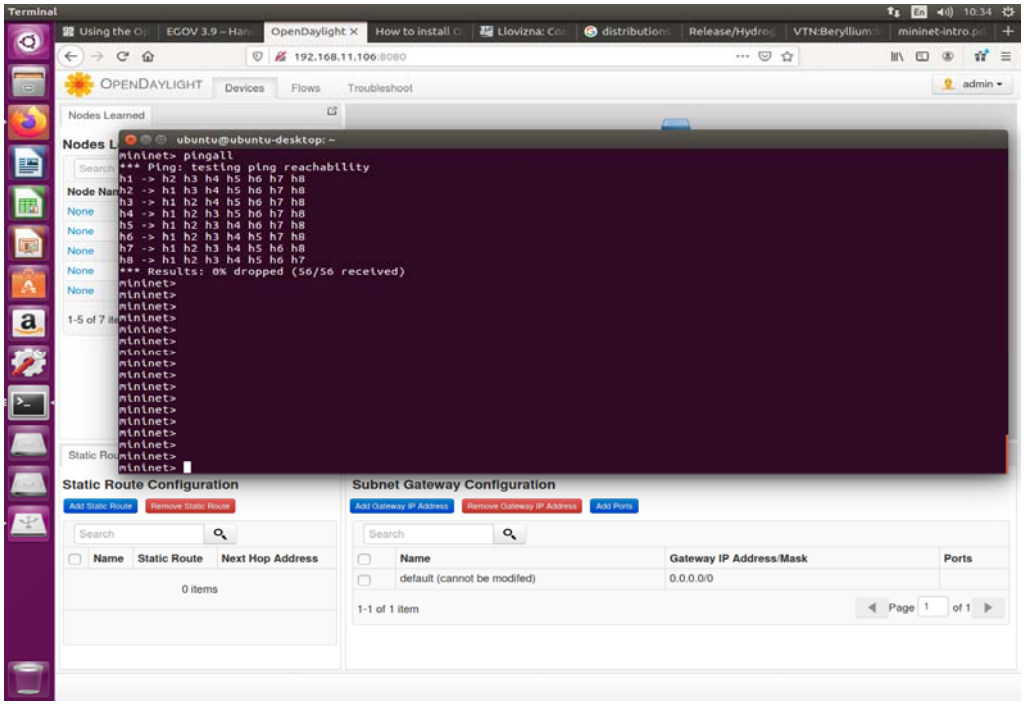
Giao diện web opendaylight



Hình 3.10: Giao diện topo tree trên Opendaylight Controller

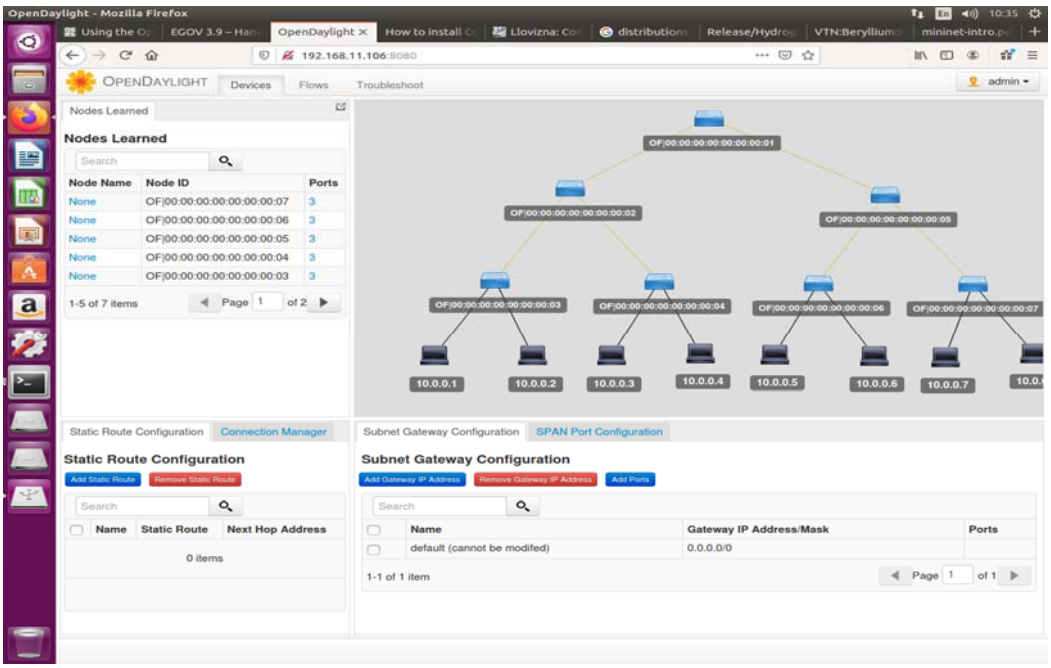
Sau khi đã load xong mô hình mạng vào mininet ta kiểm tra xem các thiết bị trong mạng đã được thông với nhau chưa.

- Khi khởi động controller ta thực hiện lệnh pingall thì tất cả gói tin đều đến đích. Điều đó nghĩa là các dưới sự điều khiển của controller, các switch và các host đã có thể liên lạc với nhau.



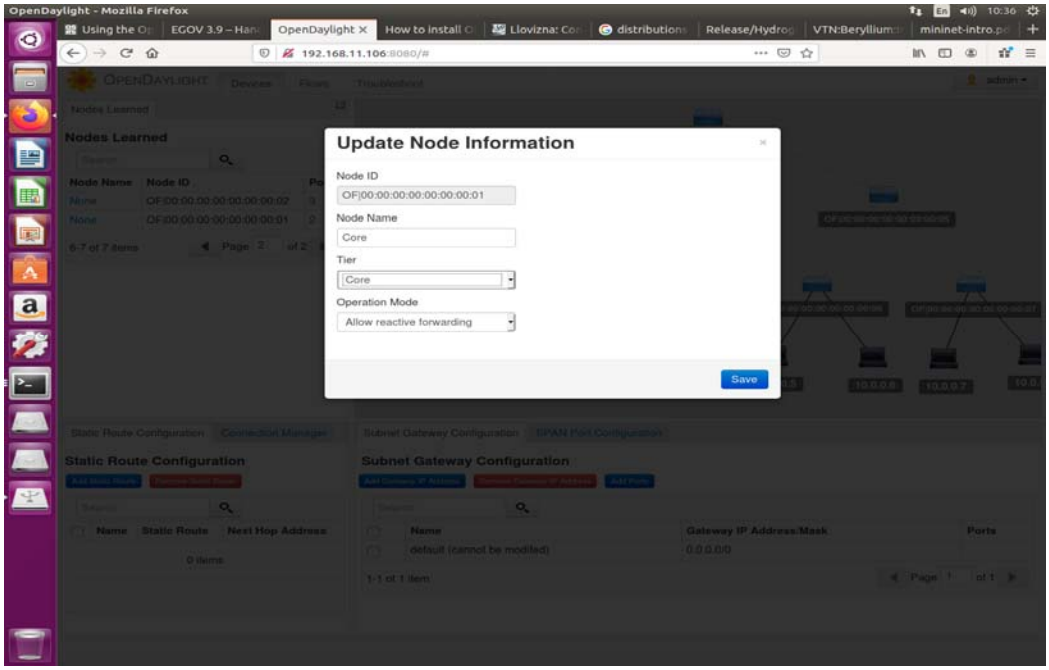
Hình 3.11: Kết quả pingall trường hợp có controller

Giao diện pingall trên opendaylight:



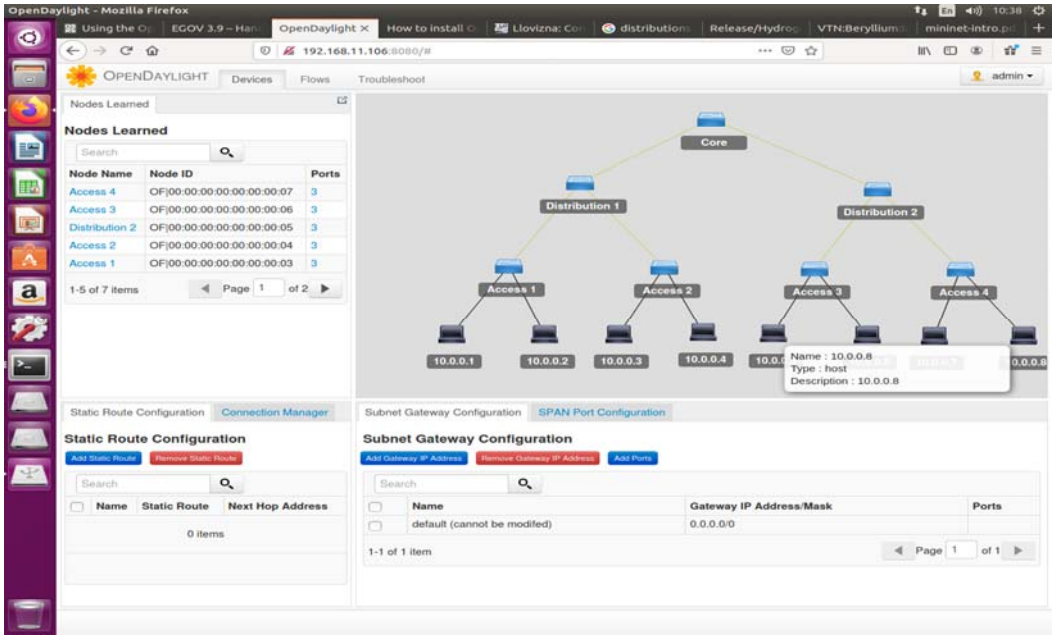
Hình 3.12: Kết quả pingall trên opendaylight controller

Đặt tên cho các node:



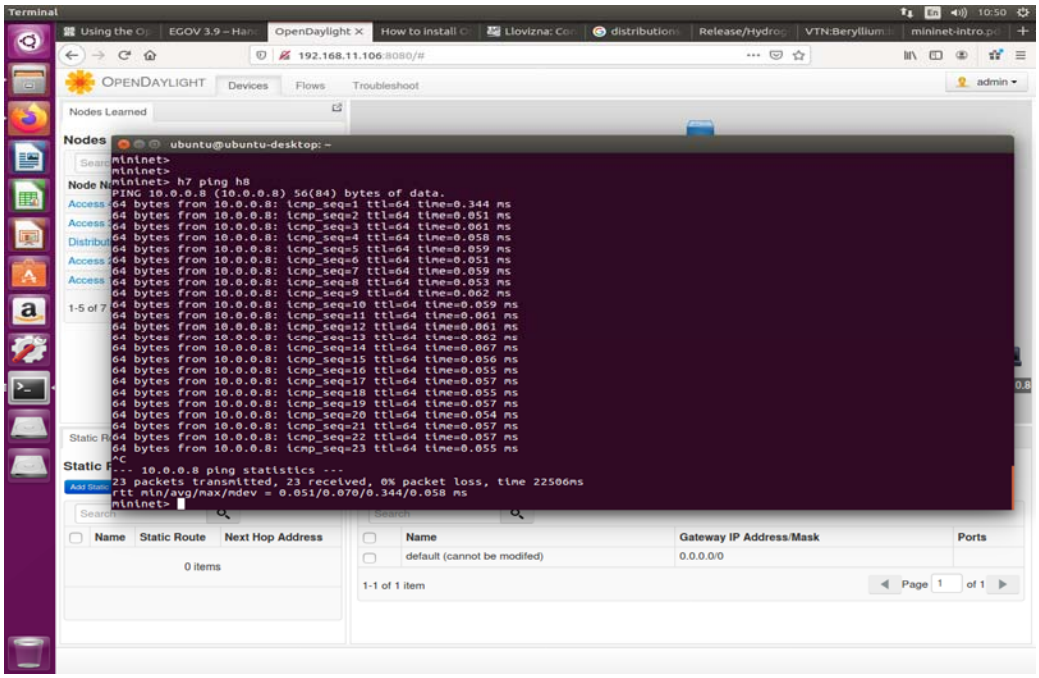
Hình 3.13: Giao diện đặt tên cho Node

Giao diện sau khi đặt tên node:



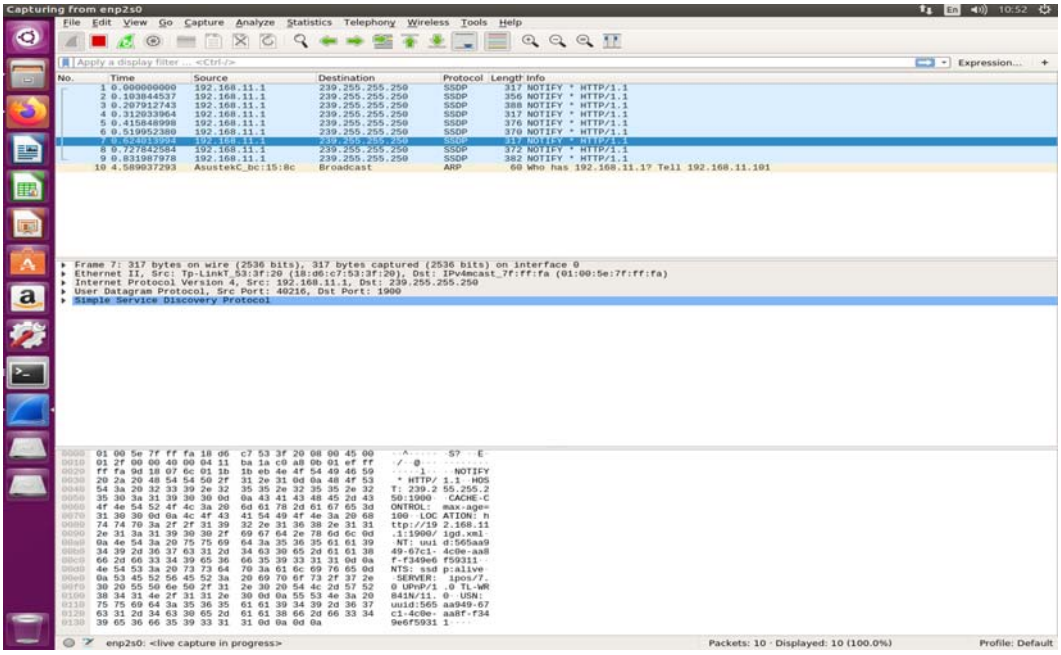
Hình 3.14: Giao diện đặt tên Node hoàn chỉnh

+ Tiến hành kiểm tra ping từ h7 đến h8.



Hình 3.15: Kết quả ping từ h7 đến h8

+ Kết quả bắt gói tin bằng wireshark:



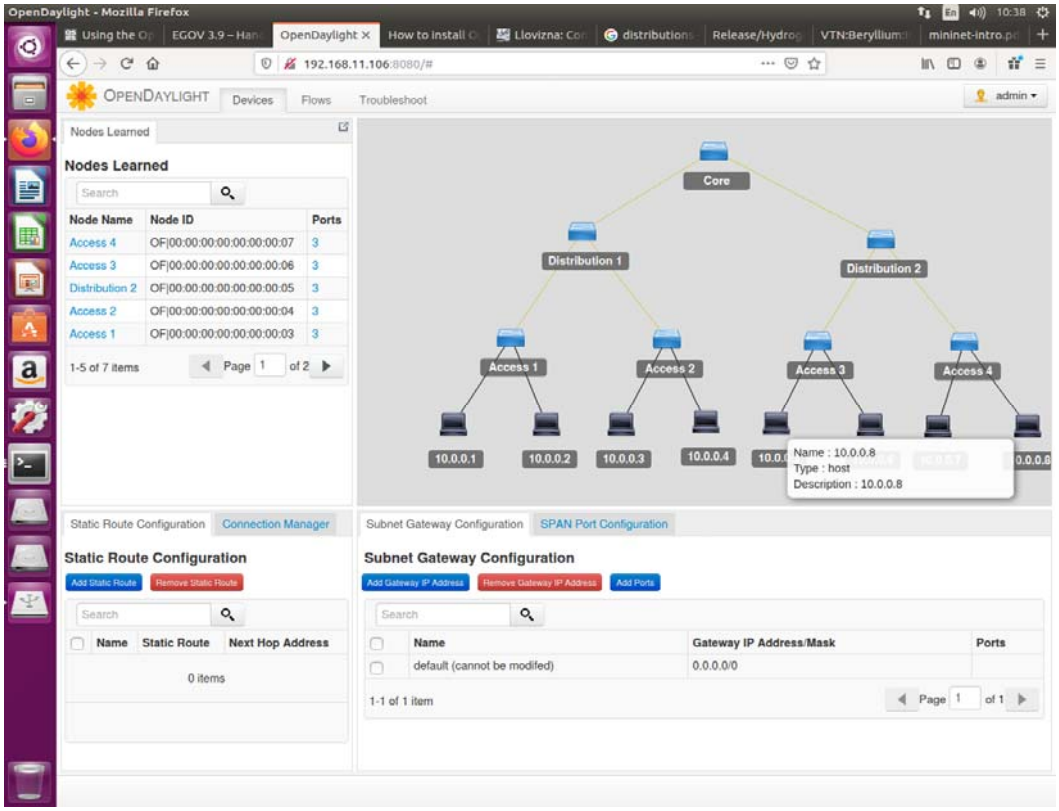
Hình 3.16: Kết quả khi bắt gói tin bằng wireshark

+ Ta thấy h7 đã kết nối được đến h8.

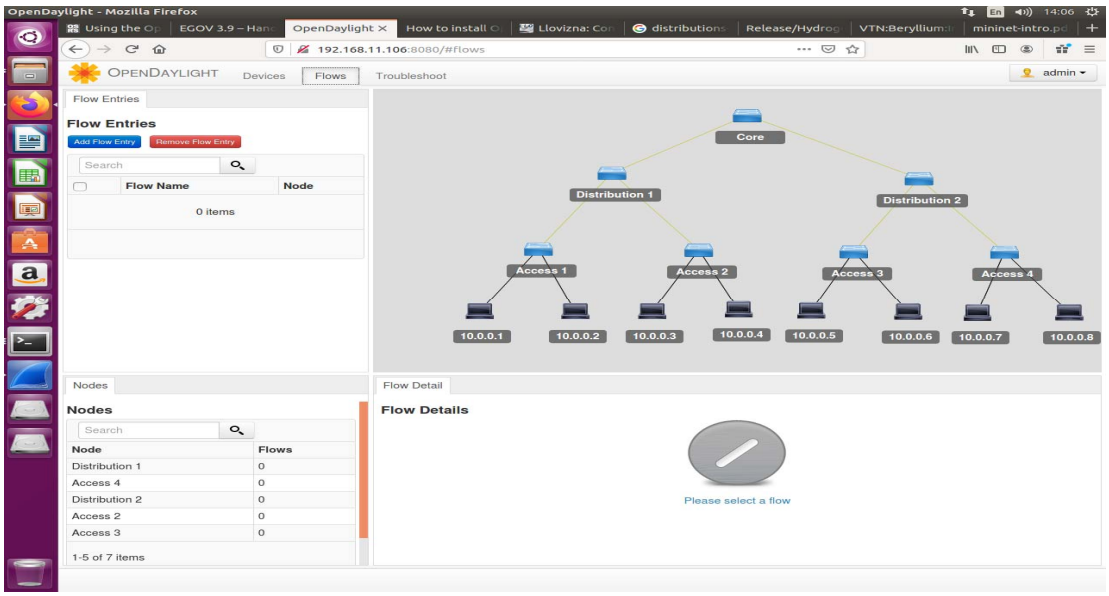
- Tiếp theo ta tiến hành truy cập vào controller để kiểm tra các thông tin cũng như có cái nhìn tổng quan hơn về chức năng của nó.

+ Vào trình duyệt gõ địa chỉ ip 192.168.11.106:8080 để truy cập vào phần điều khiển controller.

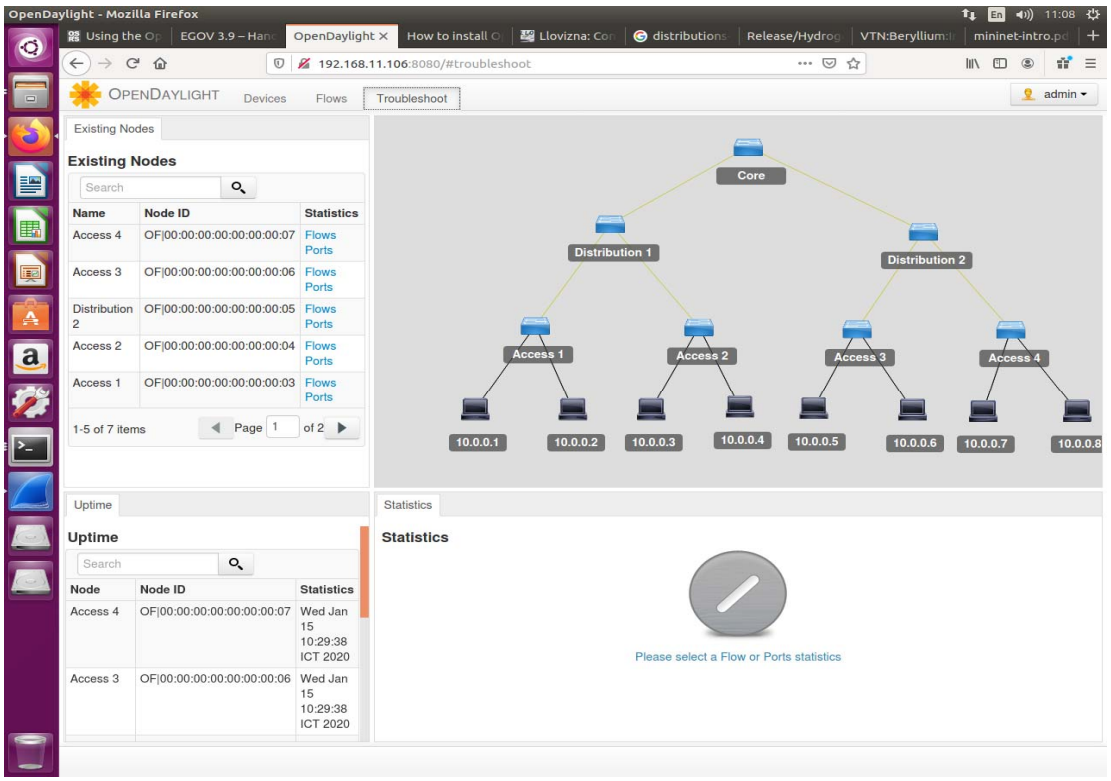
3.6. Kết quả mô phỏng



Hình 3.17a: Giao diện tab device của Opendaylight controller



Hình 3.17b: Giao diện tab flow của Opendaylight controller



Hình 3.17c: Giao diện tab Troubleshoot Opendaylight controller

+ Ta thấy sơ đồ tổng quan của mạng được thể hiện trên các hình (3.17). Đây là một trong những ưu điểm của SDN so với mạng IP truyền thống:

Với SDN người quản trị sẽ có cái nhìn tổng quan về cấu trúc mạng, có thể cấu hình các thiết bị chuyển mạch một cách dễ dàng và nhanh chóng bằng việc quản lý controller để thiết lập các flow entry để phục vụ các chức năng mạng.

Với SDN giúp người quản trị mạng phát hiện lỗi một cách nhanh chóng và dễ dàng, từ đó khắc phục nhanh nhất và hiệu quả nhất.

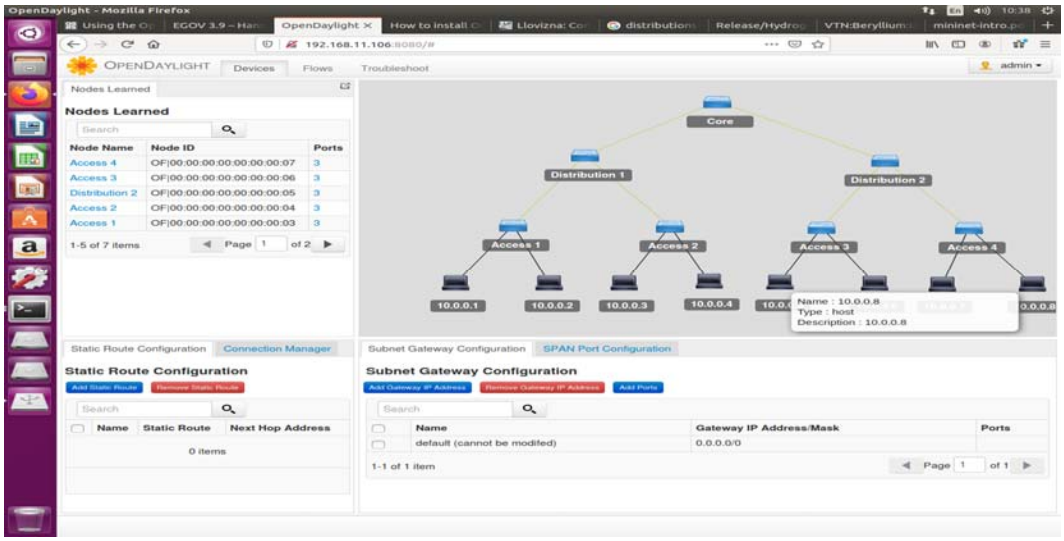
Dựa vào SDN người quản trị mạng có thể có quyền kiểm soát mạng một cách đơn giản và tập trung nên sẽ làm việc hiệu quả hơn và tránh các sự cố phát sinh.

SDN cho phép sử dụng không hạn chế và có thể thay đổi các chính sách mạng để phát hiện sự xâm nhập, tường lửa và tạo sự cân bằng với sự thay đổi của phần mềm.

SDN cho phép người quản trị dễ dàng theo dõi, phân tích lưu lượng.

Giảm chi phí đầu tư và chi phí vận hành.

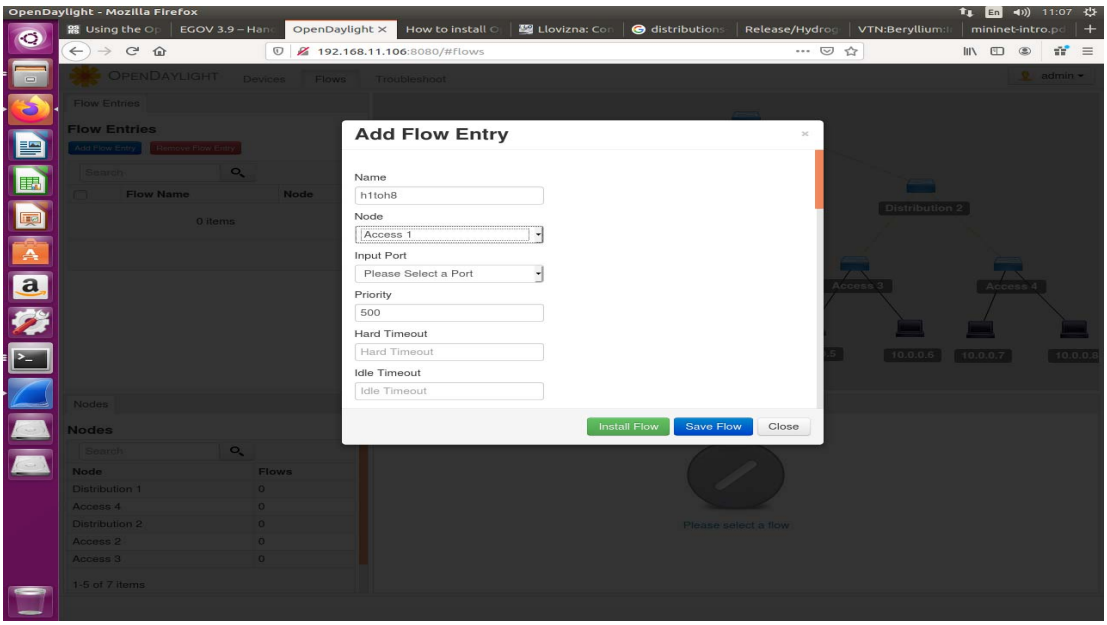
+ Ở bên trái là mục Nodes learned, nó cho ta biết các node mạng với các ID là địa chỉ MAC của chúng và các port có sự liên kết với các node. Ở phía dưới là mục Connection Manager cho phép chúng ta quản lý các kết nối của các liên kết.



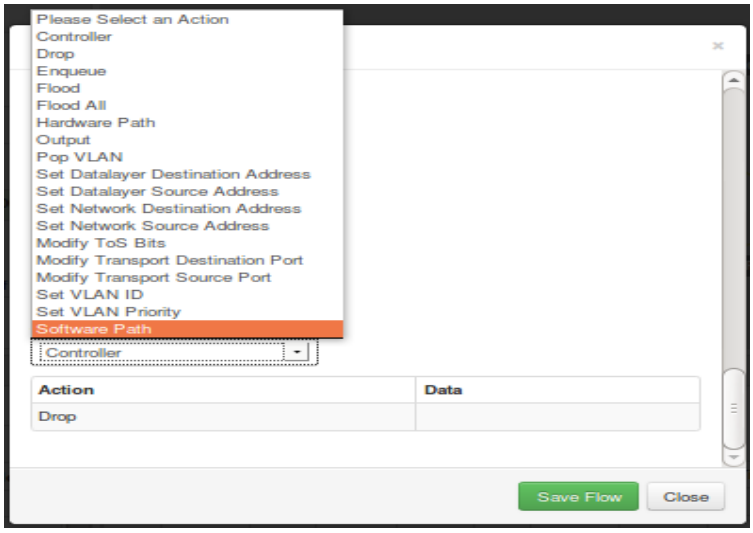
Hình 3.18: Giao diện tab flows của Opendaylight controller

+ Tab thứ 2 là tab flow, tab này cho cung cấp công cụ để quản lý các flow cũng như cho phép chúng ta điều khiển hoạt động của mạng bằng cách thêm bớt các flow tương ứng.

+ Nếu muốn thêm một Flow Entry thì ta chọn vào Add Flow Entry



Hình 3.19: Giao diện khi nhấn Add Flow Entry



Hình 3.20: Các action có thể thực hiện

+ Với các action có thể thêm vào flow entry ta có thể thực hiện quản trị mạng một cách tốt hơn.

+ Chuyển sang tab Troubleshoot, ở tab này chúng ta có thể tiến hành xem xét các flow table của các node mạng và từ đó có thể dễ dàng gỡ lỗi khi mắc phải.

3.7. Kết Luận chương

Với mô hình mạng của trường Đại học Hà Nội gồm nhiều tòa nhà được phân chia thành nhiều khu vực khác nhau nên việc ứng dụng công nghệ SDN là rất khó khăn. Chính vì vậy luận văn chỉ đưa ứng dụng SDN cho một mô hình mạng nhỏ trong hệ thống mạng toàn trường – mô hình mạng tòa nhà C. Ở chương này đã hướng dẫn cách thiết lập được một mô hình mạng trên nền tảng SDN cho mô hình mạng của một tòa nhà C của trường Đại học Hà Nội. Ta đã mô phỏng được quá trình gửi nhận các bản tin, thiết lập được các flow để thực hiện một số công việc theo nhu cầu của người sử dụng. Qua đó chúng ta có thể thấy được những ưu điểm vượt bậc của mạng SDN so với mạng IP truyền thống.

Với những ưu điểm vượt bậc so với mạng truyền thống, trong thời gian tới luận văn sẽ đưa ứng dụng SDN cho toàn bộ hệ thống mạng nội bộ trường Đại học Hà Nội để có thể tiết kiệm được một lượng lớn thời gian và nâng cao sự linh hoạt của hệ thống.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN ĐỀ TÀI

Ngày nay, Internet đã trở thành cuộc cách mạng lớn của nhân loại, kéo theo đó là sự phát triển của nhu cầu của người sử dụng, các nhu cầu đó ngày càng đa dạng và phức tạp hơn. Với cấu hình mạng truyền thống thì việc đáp ứng đó là không thể. Vì thế các nhà nghiên cứu đã phát triển và cho ra đời một cấu trúc mạng tốt hơn, đơn giản hơn và linh động hơn. SDN chính là thành quả của sự nghiên cứu đó. Nó đã giải quyết hầu như tất cả các mặt hạn chế của mạng hiện tại, cung cấp một môi trường mở cho phép các nhà phát triển tự do sáng tạo các phương thức định tuyến, phương thức bảo mật mới tốt hơn, các ứng dụng dịch vụ mới đáp ứng nhu cầu của người sử dụng.

Luận văn này đã nêu ra một cái nhìn tổng quan về mạng SDN và giao thức hỗ trợ nó - giao thức OpenFlow. Tuy chỉ áp dụng vào mô hình mạng nhỏ của một tòa nhà của trường Đại học Hà Nội nhưng nó cũng đã cho ta thấy nhưng hiệu quả đáng kể mà SDN mang lại. Do thời gian không cho phép và kiến thức còn chưa hoàn thiện nên luận văn còn nhiều điều thiếu sót và chỉ mới khai thác một phần rất nhỏ của mạng SDN. Luận văn chưa giới thiệu được các phương thức kết nối và bảo mật giữa các controller và chỉ ứng dụng được trong phạm vi nhỏ hẹp như mạng campus, chưa làm rõ được chức năng ảo hóa mạng cũng như các ứng dụng và lợi ích của việc ảo hóa mang lại.

Trong thời gian tới, dựa trên nền tảng các kết quả đã đạt được, khắc phục những thiếu sót còn lại và hướng nghiên cứu tiếp theo của luận văn là áp dụng SDN cho toàn bộ hệ thống mạng của trường Đại học Hà Nội.

TÀI LIỆU THAM KHẢO

[1] Bakshi, Kapil. "Considerations for Software Defined Networking (SDN): Approaches and use cases." Aerospace Conference, 2013 IEEE. IEEE, 2013.

[2] Nadeau, Thomas D and Ken Gray. SDN: Software Defined Networks. " O'Reilly Media, Inc.", 2013.

[3] IBM, Software Defined Networks, October 2012.

[4] Wendong, Wang, et al. "Autonomicity design in OpenFlow based Software Defined Networking." Globecom Workshops (GC Wkshps), 2012 IEEE. IEEE, 2012.

[5] Nunes, Bruno Astuto A, et al. "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks." IEEE Communications Surveys and Tutorials (Under Review).

[6] McKeown, Nick, et al. "OpenFlow: enabling innovation in campus networks."ACM SIGCOMM Computer Communication Review 38.2 (2008).

[7] Specification, OpenFlow Switch. "Version 1.1. 0 Implemented." (2011).

[8] Vivek Tiwari. SDN and Openflow for beginners with hands on labs, 2013

[9] http://mininet.org/

[10] https://wiki.opendaylight.org/

PHỤ LỤC 1

(Code mô hình mạng)

```
#!/usr/bin/python

from mininet.net import Mininet

from mininet.node import Controller, RemoteController, OVSSwitch, UserSwitch

from mininet.cli import CLI

from mininet.log import setLogLevel

from mininet.link import Link, TCLink

def topology():

    "Create a network."

    net = Mininet( controller=RemoteController, link=TCLink, switch=OVSSwitch )

    print "*** Creating nodes"

    c1 = net.addController( 'c1', controller=RemoteController, ip='192.168.56.102',
port=6633 )

    c2 = net.addController( 'c2', controller=RemoteController, ip='192.168.56.103',
port=6633 )

    s2 = net.addSwitch( 's2', listenPort=6634, mac='00:00:00:00:00:02' )

    s3 = net.addSwitch( 's3', listenPort=6635, mac='00:00:00:00:00:03' )

    s4 = net.addSwitch( 's4', listenPort=6636, mac='00:00:00:00:00:04' )

    s5 = net.addSwitch( 's5', listenPort=6637, mac='00:00:00:00:00:05' )

    s6 = net.addSwitch( 's6', listenPort=6638, mac='00:00:00:00:00:06' )

    s7 = net.addSwitch( 's7', listenPort=6639, mac='00:00:00:00:00:07' )
```

```
s8 = net.addSwitch( 's8', listenPort=66310, mac='00:00:00:00:00:08' )

s9 = net.addSwitch( 's9', listenPort=66311, mac='00:00:00:00:00:09' )

h10 = net.addHost( 'h10', mac='00:00:00:00:00:10', ip='10.0.2.10/8' )

h11 = net.addHost( 'h11', mac='00:00:00:00:00:11', ip='10.0.2.11/8' )

h12 = net.addHost( 'h12', mac='00:00:00:00:00:12', ip='10.0.2.12/8' )

h13 = net.addHost( 'h13', mac='00:00:00:00:00:13', ip='10.0.2.13/8' )

h14 = net.addHost( 'h14', mac='00:00:00:00:00:14', ip='10.0.2.14/8' )

h15 = net.addHost( 'h15', mac='00:00:00:00:00:15', ip='10.0.2.15/8' )

h16 = net.addHost( 'h16', mac='00:00:00:00:00:16', ip='10.0.2.16/8' )

h17 = net.addHost( 'h17', mac='00:00:00:00:00:17', ip='10.0.2.17/8' )

print "*** Creating links"

net.addLink(h17, s9, 0, 2)

net.addLink(h16, s8, 0, 2)

net.addLink(h15, s7, 0, 2)

net.addLink(h14, s6, 0, 2)

net.addLink(h13, s5, 0, 2)

net.addLink(h12, s4, 0, 2)

net.addLink(h11, s3, 0, 3)

net.addLink(h10, s3, 0, 2)

net.addLink(s2, s9, 7, 1)

net.addLink(s2, s8, 6, 1)

net.addLink(s2, s7, 5, 1)
```

```
    net.addLink(s2, s6, 4, 1)
    net.addLink(s2, s5, 3, 1)
    net.addLink(s2, s4, 2, 1)
    net.addLink(s2, s3, 1, 1)
    print "*** Starting network"
    net.start()
    c1.start()
    c2.start()
    s2.start( [c1,c2] )
    s3.start( [c1,c2] )
    s4.start( [c1,c2] )
    s5.start( [c1,c2] )
    s6.start( [c1,c2] )
    s7.start( [c1,c2] )
    s8.start( [c1,c2] )
    s9.start( [c1,c2] )
    net.staticArp()
    print "*** Running CLI"
    CLI( net )
    print "*** Stopping network"
    net.stop()
if __name__ == '__main__':
```

```
setLogLevel( 'info' )

topology()
```