BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



BÀI BÁO CÁO THỰC TẬP CƠ SỞ

CHỦ ĐỀ

TÌM HIỂU VÀ MÔ PHỎNG MẠNG SDN SỬ DỤNG MININET (ODL)

**Giảng viên hướng dẫn : Cấn Thị Phượng**

**Sinh viên thực hiện : Hàng Minh Sơn**

**Mã số sinh viên : 61134291**

KHÁNH HÒA - 2021

**LỜI CAM ĐOAN**

Tôi xin được cam đoan : Đề tài báo cáo thực tập cơ sở “Tim hiểu và mô phỏng SDN sử dụng mininet (ODL)” là kết quả dựa trên sự cô gắng, nỗ lực của bản thân với sự hướng dẫn nhiệt tình của giảng viên Cấn Thị Phượng. Các kết quả và nghiên cứu trong đề tài là trung thực và hoàn toàn không sao chép hoặc sử dụng kết quả của đề tài nghiên cứu nào tương tự.

Những phần sử dụng tài liệu tham khảo trong đồ án đã trích dẫn đầy đủ.

Nếu phát hiện có sự sao chép kết quả nghiên cứu của đề tài khác, tôi xin chịu hoàn toàn trách nhiệm và kỷ luật của giảng viên đưa ra.

*Khánh Hòa, ngày 30 tháng 12 năm 2021*

**Sinh viên thực hiện**

Hàng Minh Sơn

**LỜI CẢM ƠN**

Để hoàn thành đề tài báo cáo thực tập cơ sở này, trước hết em xin gửi đến quý thầy, cô Khoa Công nghệ Thông tin – Trường Đại Học Nha Trang lời cảm ơn chân thành.

Em xin gửi đến cô Cấn Thị Phượng, người đã tận tình hướng dẫn, giúp đỡ em hoàn thành đề tài báo cáo này lời cảm ơn sâu sắc nhất.

Trong quá trình nghiên cứu thực hiện đề tài, cũng như trong quá trình làm bài báo cáo thực tập cơ sở, khó tránh khỏi sai sót, rất mong các thầy, cô bỏ qua. Đồng thời do kiến thức cũng như kinh nghiệm thực tiễn của bản thân còn hạn chế, bài báo cáo này khó tránh khỏi những thiếu sót, em rất mong nhận được ý kiến đóng góp từ quý thầy, cô để em học hỏi thêm được nhiều kinh nghiệm, cũng như kỹ năng cần thiết.

Em xin chân thành cảm ơn!

LỜI MỞ ĐẦU

Internet ngày càng phát triển và quan trọng đối với mỗi chúng ta. Sự phát triển hàng ngày, hàng giờ với các tính năng mới mang đến cho người dùng những trải nghiệm và phục vụ tốt hơn nhu cầu cuộc sống mỗi người.

đi xuống một cấp độ thấp hơn, cấp độ mạng, chúng ta có thể nhận ra rằng, sự phát triển ở cấp độ này diễn ra chậm hơn rất nhiều. Không có nghi ngờ nào về sự phát triển ngày càng mạnh mẽ của cơ sở hạ tầng mạng internet trên “mặt số lượng”, các kỹ thuật mới ở Layer 2 được áp dụng, tuy nhiên sự thay đổi về mặt cấu trúc đến thời điểm hiện tại là gần như không đáng kể. Một trong những nguyên nhân cho vấn đề này là vì cấu trúc mạng “ nguyên khối “, nó chứa tập hợp các chức năng trong đó kể cả các ứng dụng mang. Việc áp dụng chức năng mới yêu cầu phải hiện đại hóa toàn mạng với hàng triệu thiết bị. Hãy thử tưởng tượng rằng chúng ta phải tiến hành cập nhật tất cả các thiết bị mỗi khi chúng ta cài một ứng dụng mới, đó thực sự là một công việc khó khăn và mất rất nhiều thời gian, công sức.

Nói cách khác, sự đổi mới trên cấp độ mạng trong khuôn khổ cấu trúc ngày nay là rất khó khăn. Các chức năng và các tính năng mới làm tăng tính phức tạp của hệ thống lên rất nhiều lần, việc thử nghiệm chúng cũng vậy và nếu áp dụng chúng vào thực tế cũng đòi hỏi chi phí rất cao và tiềm ẩn nhiều nguy cơ khác. Chính vì thế rất nhiều chuyên gia đã đặt kỳ vọng vào một mô hình mạng mới, mạng điều khiển bởi phần mềm SDN. Bài báo cáo cho chúng ta thấy một cách tổng quan về mạng SDN.

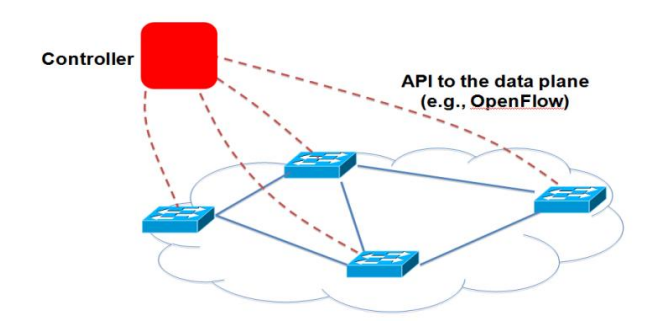
Vì những lí do trên nên em viết định chọn đề tài “ **Tìm hiểu và mô phỏng mạng sử dụng mininet (ODL)**” để báo cáo.

Do kiến thức và thời gian có hạn, không tránh khỏi sai sót, kính mong các thầy, cô góp ý kiến để bài báo cáo hoàn thiện hơn.

Chương 1: tổng quan về SDN

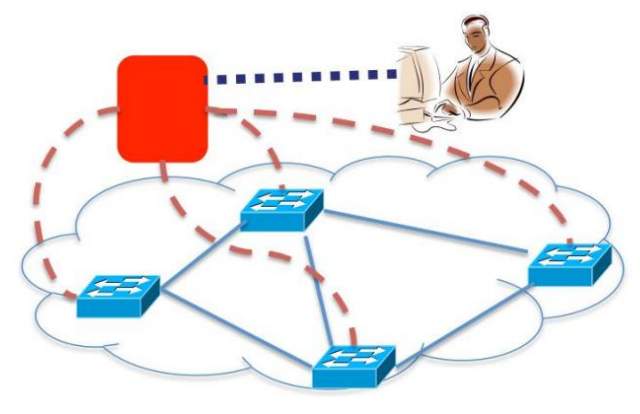
1.1 Giới thiệu SDN

SDN là một cách tiệp cận để quản lý mạng cho phép cấu hình mạng động, hiệu quả theo lập trình để cải thiện hiệu suất và giám sát mạng, làm cho nó giống như điện toán đám mây hơn là quản lý mạng truyền thông. SDN có nghĩa là giải quyết kiến trúc tĩnh của các mạng truyền thông.



*Hình 1. Controller quản lý tập trung các thiết bị switch thông qua API*

Trong SDN, control plane được tách ra từ các thiết bị vật lý và chuyển đến các controller. Controller này có thể nhìn thấy toàn bộ mạng và do đó cho phép các kỹ sư mạng làm cho chính sách chuyển tiếp được tối ưu dựa trên toàn bộ mạng. Các controller tương tác với các thiết bị mạng vật lý thông qua một giao thức OpenFlow.

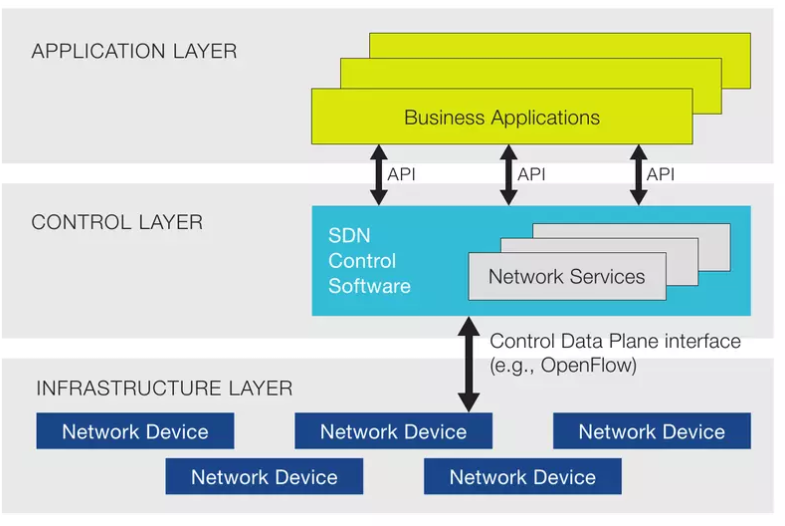


*Hình 2. Quản lý tập trung thông qua controller duy nhất*

1.2 Kiến trúc của SDN

Về cơ bản kiến trúc của SDN bao gồm ba lớp: lớp ứng dụng (Appication Layer), lớp điều khiển (Control Layer) và lớp hạ tầng (Infrastructure Layer).

* Lớp ứng dụng: là các ứng dụng kinh doanh được triển khai trên mạng, được kết nối tới lớp điều khiển thông qua các API, cung cấp khả năng cho phép lớp ứng dụng lập trình lại (cấu hình lại) mạng (điều chỉnh các tham số trễ, băng thông, định tuyến, …) thông qua lớp điều khiển.
* Lớp điều khiển: là nơi tập trung các bộ điều khiển thực hiện việc điều khiển cấu hình mạng theo các yêu cầu từ lớp ứng dụng và khả năng của mạng.
* Lớp cơ sở hạ tầng: là các thiết bị mạng thực tế (vật lý hay ảo hóa) thực hiện việc chuyển tiếp gói tin theo sự điều khiển của lớp điểu khiển.



*Hình 3. Kiến trúc SDN*

1.3 Ưu - nhược điểm SDN

1.3.1 Ưu điểm

• Controller có thể được lập trình trực tiếp.

• Cấu hình lớp cơ sở hạ tầng có thể được lập trình trên lớp ứng dụng và truyền đạt xuống các lớp dưới.

• Giảm CapEx: SDN giúp giảm thiểu các yêu cầu mua phần cứng theo mục đích xây dựng các dịch vụ để loại bỏ lãng phí cho việc dự phòng.

• Giảm OpEx: thông qua các phần tử mạng đã được gia tăng khả năng lập trình, 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 tới việc tối ưu khả năng và độ tin cậy của dịch vụ.

1.3.2 Nhược điểm

* Độ trễ **:** Mọi thiết bị được sử dụng trên mạng đều chiếm một không gian trên đó. Tốc độ tương tác giữa các thiết bị và mạng phụ thuộc vào số lượng tài nguyên ảo hóa. Nếu cần tốc độ cao hơn, nhiều tài nguyên ảo hóa hơn có thể được giới thiệu. Giờ đây, việc ảo hóa tài nguyên có thể dẫn đến độ trễ đáng kể.
* Bảo trì **:** Bảo trì là một khía cạnh rất quan trọng của mạng để thực hiện các hoạt động của nó. SDN thiếu ở khía cạnh bảo trì. Nó làm cho hầu như không thể quản lý các thiết bị thực tế. Đặc biệt là trong khi nâng cấp mạng.
* Sự phức tạp **:** Không có bất kỳ giao thức bảo mật tiêu chuẩn nào cho SDN. Mặc dù có một số nhà cung cấp dịch vụ bên thứ ba, vẫn tồn tại những lo ngại về bảo mật. Chỉ những người có chuyên môn trong việc xử lý các hệ thống SDN mới có thể ngăn chặn các cuộc tấn công lớn.
* Cấu hình **:** Cấu hình lại mạng SDN không phải là một nhiệm vụ đơn giản vì nó liên quan đến rất nhiều chi phí. Đặc biệt, việc triển khai các giao thức SDN và bộ điều khiển không thể được thực hiện bằng cách cấu hình từng giao thức. Toàn bộ mạng cần được cấu hình lại.
* Bảo mật thiết bị **:** SDN không sử dụng các bộ định tuyến và thiết bị chuyển mạch thông thường. Do đó, bảo mật đi kèm với chúng bị bỏ qua. Ví dụ: bảo mật đi kèm với tường lửa không có trong SDN. Điều này có nghĩa là mạng của bạn dễ bị tấn công hơn trước các mối đe dọa từ bên ngoài.

1.5 Ứng dụng SDN

* DevOps – phương pháp tiếp cận dựa trên SDN có thể giúp tự động hóa các bản cập nhật và triển khai ứng dụng, từ đó tạo điều kiện thuận lợi cho DevOps. Cụ thể, SDN giúp tự động hóa các thành phần có sở hạ tầng khi các ứng dụng và nền tảng của DevOps được triển khai.
* Campus network (mạng campus) – Các mạng campus thường khó để quản lý, đặc biệt là khi đang có nhu cầu về việc thống nhất mạng WiFi và Ethernet. SDN controller có thể giúp cung cấp khả năng quản lý tập trung và tự động hóa cho mạng campus. Từ đó cải thiện bảo mật và chất lượng dịch vụ ở cấp độ ứng dụng trên toàn mạng.
* Mạng nhà cung cấp dịch vụ – SDN giúp các nhà cung cấp dịch vụ đơn giản hóa và tự động hóa việc cung cấp mạng để quản lý và kiểm soát dịch vụ và end-to-end.

Chương 2 : Tìm hiểu OpenFlow

2.1 Giới thiệu OPENFLOW

OpenFlow là một giao thức truyền thông cho phép truy cập vào mặt phẳng chuyển tiếp của bộ chuyển đổi mạng hoặc bộ định tuyến qua mạng.OpenFlow cho phép truy cập trực tiếp và điều khiển mặt phẳng chuyển tiếp của các thiết bị mạng như switch và router, cả thiết bị vật lý và thiết bị ảo, do đó giúp di chuyển phần điều khiển mạng ra khỏi các switch thực tế tới phần mềm điều khiển trung tâm.

Sự xuất hiện của OpenFlow thực sự là một cuộc cách mạng, đưa sự phát triển của SDN lên một tầm cao mới. OpenFlow là giao thức hoạt động giữa tầng điều khiển (Control Layer) và tầng vật lý (Infrastructure Layer). Trong kiến trúc của SDN, tất cả các thiết bị được liên kết với tầng điều khiển và thông qua OpenFlow. OpenFlow có 2 nhiệm vụ chính:

• Giám sát hoạt động của các thiết bị mạng: Lưu lương mạng, trạng thái hoạt động của các nút mạng, các thông tin cơ bản về các thiết bị …

• Điều khiển hoạt động của thiết bị mạng: Điều khiển luồng dữ liệu (routing), Bảo mật, Quality of Service...

2.2 Các đặc trưng của openflow

+ OpenFlow có thể được sử dụng bởi ứng dụng phần mềm ngoài để điều khiển mặt phẳng 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 kết nối giữa các thiết bị cơ sở hạ tầng mạng và phần mềm điều khiển SDN.

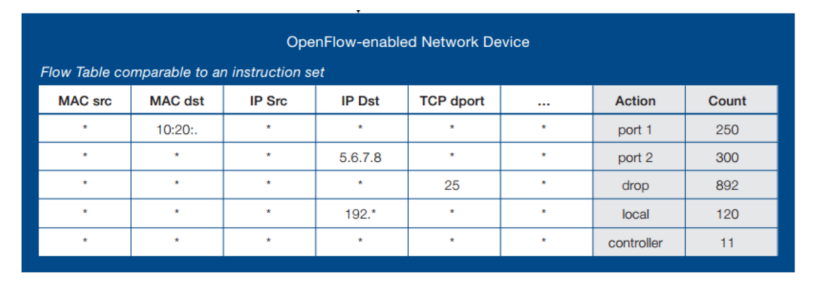
+OpenFlow sử dụng khái niệm “flow” (luồng) để nhận dạng lưu lượng mạng trên cơ sở định nghĩa trước các quy tắc phù hợp (được lập trình sẵn bởi phần mềm điều khiển SDN). Giao thức này còn cho phép định nghĩa các tham số, chẳng hạn như mô hình lưu lượng sử dụng, ứng dụng và tài nguyên cần thiết để xác định cách thức điều hướng lưu lượng truyền qua các thiết bị mạng.

+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.

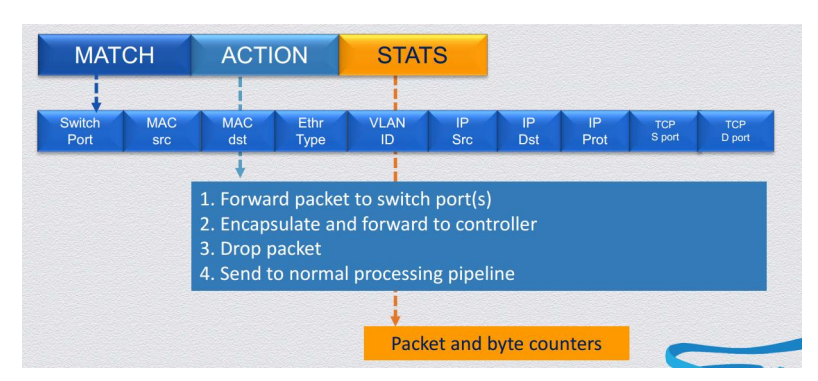
2.3 Cấu tạo và hoạt động của openflow

2.3.1 Cấu tạo

Một thiết bị OpenFlow bao gồm ít nhất 3 thành phần :



*Hình 6. Một ví dụ về Flow Table*

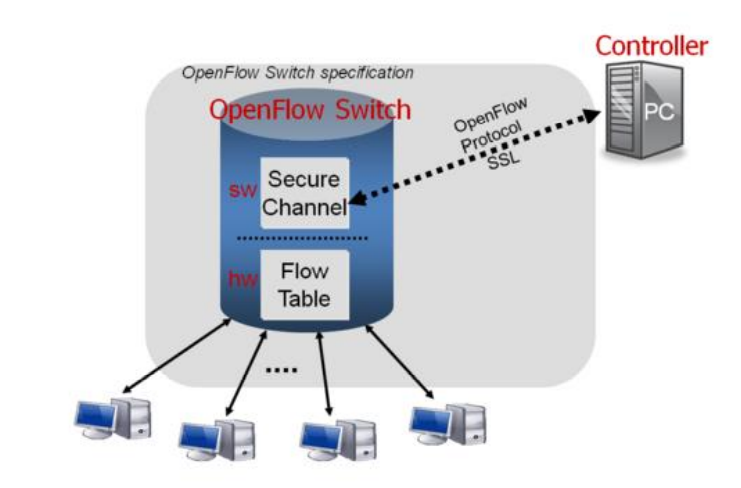
**

*Hình 7. Các trường trong bảng flow và cách thức hoạt động*

• Flow Table: một liên kết hành động với mỗi luồng, giúp thiết bị xử lý các luồng thế nào.

• Secure Channel: kênh kết nối thiết bị tới controller (controller), cho phép các lệnh và các gói tin được gửi giữa controller và thiết bị

• OpenFlow Protocol: giao thức cung cấp phương thức tiêu chuẩn và mở cho một controller truyền thông với thiết bị.



Hình 8. Các thành phần trong thiết bị OpenFlow

2.3.3 Hoạt động

Trong router hay switch cổ điển, các gói tin chuyển tiếp nhanh chóng (data path) và các quyết định định tuyến mức độ cao (control path) xảy ra trên cùng một thiết bị. OpenFlow Switch tách hai chức năng đó ra. Phần data patth vẫn nằm trên switch, trong khi control path được chuyển đến một controller riêng biệt, thường là một máy chủ tiêu chuẩn. OpenFlow Switch và Controller giao tiếp thông qua giao thức OpenFlow, trong đó xác định các tin nhắn, chẳng hạn như gói nhận được, gửi gói tin ra, sửa đổi bảng chuyển tiếp và nhận số liệu thống kê.

Các data path của một OpenFlow Switch được trình bày rõ ràng trong flow table; mỗi mục flow table chứa một tập các trường gói tin phù hợp và một hành động (như gửi ra cổng, sửa đổi trường hoặc hủy). Khi một OpenFlow Switch nhận được một gói tin nó chưa bao giờ thấy trước đây mà không có trong flow table, nó sẽ gửi gói tin này đến bộ điều khiển. Controller sau đó đưa ra quyết định về cách xử lý gói tin này. Nó có thể hủy (drop) các gói tin, hoặc nó có thể thêm một flow entry chỉ đạo việc chuyển đổi trên làm thế nào để chuyển tiếp các gói tin tương tự trong tương lai

2.3.4 LỢI ÍCH KHI SỬ DỤNG OPENFLOW

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: phần mềm điều khiển SDN có thể điều khiển bất kỳ thiết bị mạng nào cho phép OpenFlow từ 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 một framework quản lý mạng tự động và linh hoạt. Từ framework này có thể phát triển các công cụ tự động hóa các nhiệm vụ hiện đang được thực hiện bằng tay.

• Tốc độ đổi mới cao hơn: việc áp dụng OpenFlow cho phép các nhà khai thác mạng lập trình lại mạng trong thời gian thực để đạt được các nhu cầu kinh doanh và yêu cầu từ người dùng cụ thể khi có sự thay đổi.

• Gia tăng độ tin cậy và khả năng an ninh của mạng: các nhân viên IT có thể định nghĩa các trạng thái cấu hình và chính sách ở mức cao và áp dụng tới cơ sở hạ tầng thông qua OpenFlow. Kiến trúc SDN trên cơ sở OpenFlow cung cấp điều khiển và tầm nhìn hoàn chỉnh trên mạng, nên có thể đảm bảo điều khiển truy nhập, định hình lưu lượng, QoS, an ninh và các chính sách khác được thực thi nhất quán trên toàn bộ cơ sở hạ tầng mạng không dây và có dây, bao gồm cả các văn phòng chi nhánh, các cơ sở chính và DC.

• Điều khiển mạng chi tiết hơn: mô hình điều khiển trên cơ sở flow của OpenFlow cho phép nhân viên IT áp dụng các chính sách tại mức chi tiết, bao gồm phiên, người dùng, thiết bị và các mức ứng dụng trong một sự trừu tượng hóa cao, tự động điều chỉnh thích hợp. - Tốt hơn với trải nghiệm người dùng: 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.

CHƯƠNG 3 : Tổng quan về mininet

3.1 giới thiệu về mininet

Mininet là một *trình giả lập mạng,*hoặc có lẽ chính xác hơn là một *hệ thống dàn dựng mô phỏng mạng*. Nó chạy một bộ sưu tập các máy chủ cuối, công tắc, bộ định tuyến và liên kết trên một nhân Linux duy nhất. Nó sử dụng ảo hóa nhẹ để làm cho một hệ thống duy nhất trông giống như một mạng hoàn chỉnh, chạy cùng một hạt nhân, hệ thống và mã người dùng. Một máy chủ Mininet hoạt động giống như một cỗ máy thực sự; bạn có thể vào nó (nếu bạn khởi động và kết nối mạng với máy chủ của bạn) và chạy các chương trình tùy ý (bao gồm bất cứ điều gì được cài đặt trên hệ thống Linux cơ bản.) Các chương trình bạn chạy có thể gửi các gói thông qua những gì có vẻ như một giao diện Ethernet thực sự, với tốc độ liên kết nhất định và độ trễ. Các gói được xử lý bởi những gì trông giống như một công tắc Ethernet thực sự, bộ định tuyến hoặc hộp giữa, với một số lượng nhất định của hàng đợi. Khi hai chương trình, như máy khách và máy chủ, giao tiếp thông qua Mininet, hiệu suất đo được sẽ phù hợp với hai máy gốc (chậm hơn).

Nói tóm lại, máy chủ ảo, công tắc, liên kết và bộ điều khiển ảo của Mininet là điều thực sự - chúng chỉ được tạo ra bằng phần mềm chứ không phải phần cứng - và phần lớn hành vi của chúng tương tự như các yếu tố phần cứng rời rạc. Thông thường có thể tạo ra một mạng Mininet giống như một mạng phần cứng hoặc một mạng phần cứng giống như mạng Mininet và chạy cùng một mã nhị phân và các ứng dụng trên một trong hai nền tảng.

3.2 một số chức năng của mininet

Cung cấp một network testbed đơn giản và rẻ (do không tốn kém chi phí mua các switch Openflow) để phát triển các ứng dụng mạng. Do các switch Openflow trong Mininet có tất cả các tính chất mà switch OpenFlow thật có được nên việc sử dụng mạng emulator bằng Mininet là 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 đồ hình.

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

• Cho phép debug và chạy các test của các mạng lớn, sử dụng CLI.

• Hỗ trợ thiết lập các đồ hình tùy biến bất kỳ, gồm tập cơ bản các thông số đồ 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.

Mininet cho ta một phương pháp dễ dàng để thu được chính xác các đặc tính mạng và thử nghiệm các đồ hình tùy ý. Quan trọng nhất, code mà chúng ta phát triển và kiểm tra bằng Mininet cho Openflow controller, switch hay host, có thể đưa ra triển khai với hệ thống phần cứng thật không cần thay đổi.

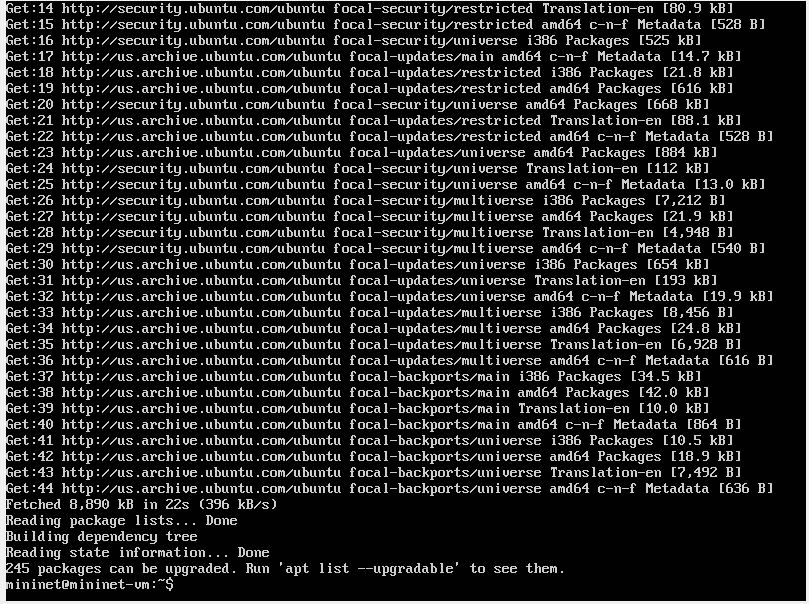
3.3 những hạn chế của mininet

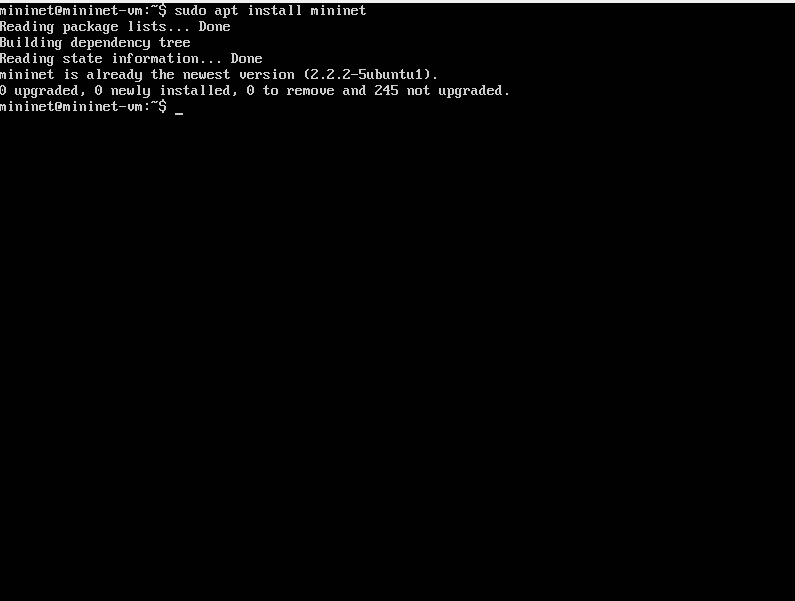
* Chạy trên một hệ thống duy nhất là thuận tiện, nhưng nó áp đặt giới hạn tài nguyên: nếu máy chủ của bạn có CPU 3 GHz và có thể chuyển đổi khoảng 10 Gbps lưu lượng mô phỏng, những tài nguyên đó sẽ cần phải được cân bằng và chia sẻ giữa các máy chủ ảo và công tắc của bạn.
* Mininet sử dụng một nhân Linux duy nhất cho tất cả các máy chủ ảo; điều này có nghĩa là bạn không thể chạy phần mềm phụ thuộc vào BSD, Windows hoặc các hạt nhân hệ điều hành khác. (Mặc dù bạn có thể đính kèm VM vào Mininet.)
* Mininet sử dụng một nhân Linux duy nhất cho tất cả các máy chủ ảo; điều này có nghĩa là bạn không thể chạy phần mềm phụ thuộc vào BSD, Windows hoặc các hạt nhân hệ điều hành khác. (Mặc dù bạn có thể đính kèm VM vào Mininet.)
* Theo mặc định, mạng Mininet của bạn bị cô lập khỏi mạng LAN của bạn và từ internet - đây thường là một điều tốt! Tuy nhiên, bạn có thể sử dụng đối tượng và/hoặc tùy chọn kết nối mạng Mininet của bạn với mạng LAN của bạn thông qua Dịch Địa chỉ Mạng. Bạn cũng có thể đính kèm giao diện phần cứng thực (hoặc ảo) vào mạng Mininet của mình (xem chi tiết.)NAT--natexamples/hwintf.py
* Theo mặc định tất cả các máy chủ Mininet chia sẻ hệ thống tệp máy chủ và không gian PID; điều này có nghĩa là bạn có thể phải cẩn thận nếu bạn đang chạy daemons yêu cầu cấu hình trong / vv, và bạn cần phải cẩn thận rằng bạn không giết sai quy trình do nhầm lẫn. (Lưu ý ví dụ chứng minh cách có thư mục riêng cho mỗi máy chủ.)bind.py

3.3 cài đặt mininet

$sudo apt update

$sudo apt install mininet



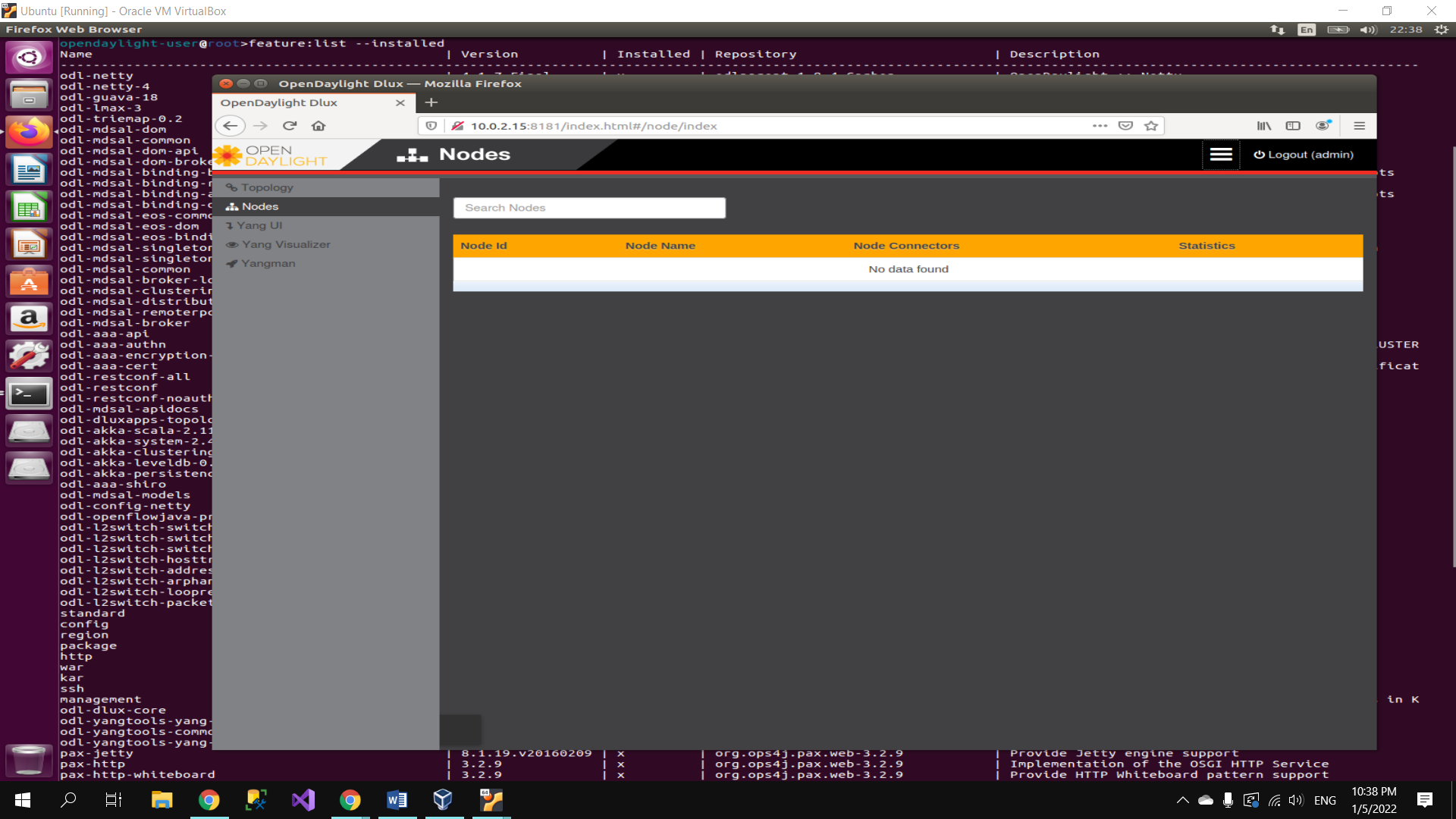


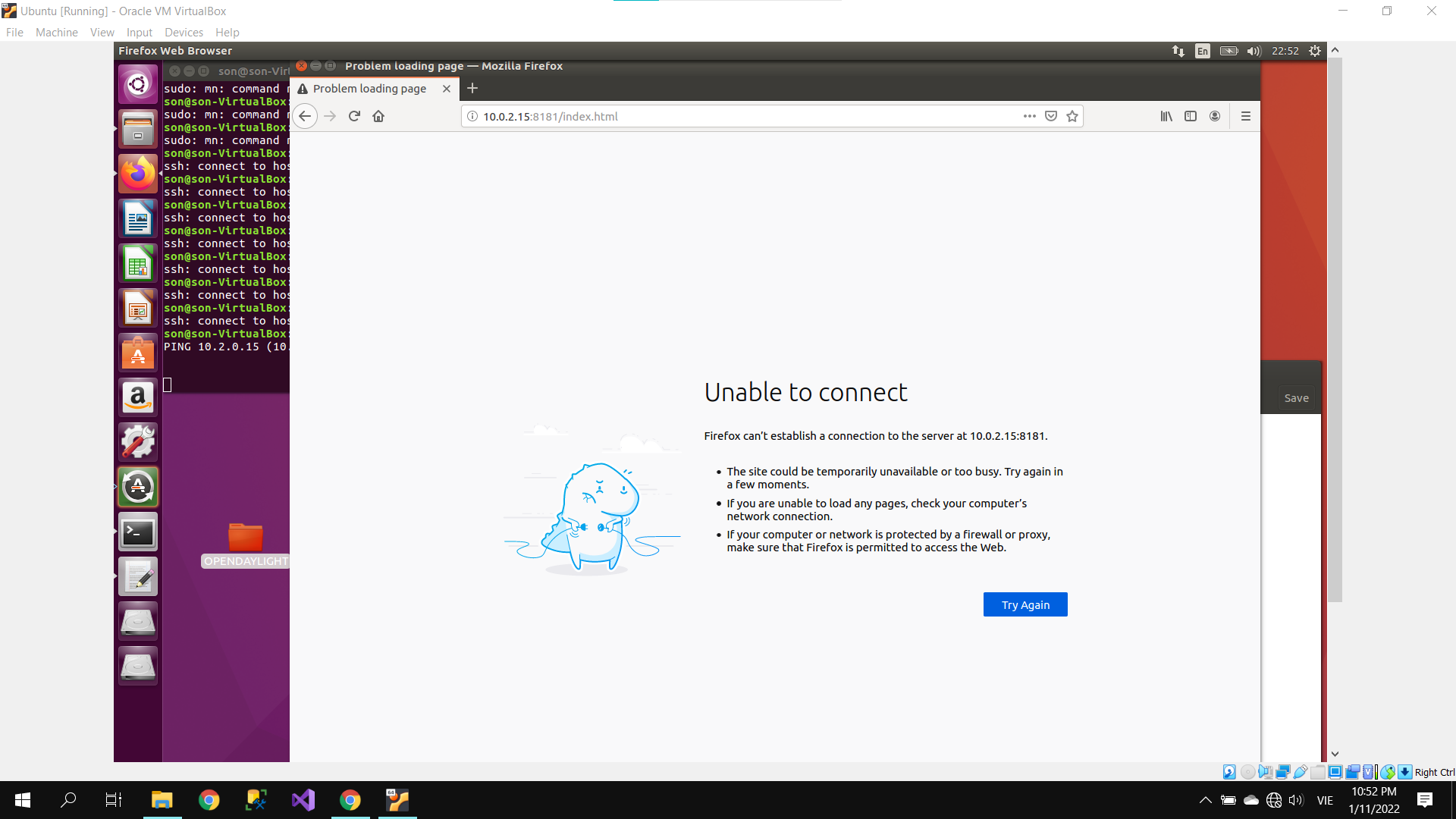
CHƯƠNG 4 : OPENDAYLIGHT và mô phỏng

4.1 giới thiệu

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

OpenDaylight là dự án mã nguồn mở hợp tác với Linux, với mục tiêu trở thành một thành phần cốt lõi của kiến trúc SDN, cho phép người sử dụng làm giảm sự phưc 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 Openflow.



[](https://www.academia.edu/9426394/SOFTWARE_DEFINED_NETWORKING_C%C3%94NG_NGH%E1%BB%86_M%E1%BB%9AI_L%C3%80M_THAY_%C4%90%E1%BB%94I_C%E1%BA%A4U_TR%C3%9AC_M%E1%BA%A0NG)

https://www.academia.edu/9426394/SOFTWARE\_DEFINED\_NETWORKING\_C%C3%94NG\_NGH%E1%BB%86\_M%E1%BB%9AI\_L%C3%80M\_THAY\_%C4%90%E1%BB%94I\_C%E1%BA%A4U\_TR%C3%9AC\_M%E1%BA%A0NG