# TRƯỜNG ĐẠI HỌC NHA TRANG KHOA CÔNG NGHỆ THÔNG TIN

o0o



**BÁO CÁO**

**Tìm hiểu và mô phỏng mạng SDN sử dụng mininet (ONOS)**

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

**Sinh viên thực hiện: Nguyễn Văn Phi**

**Mã số sinh viên: 61134168 Khóa: 61**

**Lớp: 61CNTT-3**

**MỤC LỤC**

[Chương 1: Tổng quan về SDN 3](#_Toc90647348)

[**1.1** **Giới thiệu chung** 3](#_Toc90647349)

[**1.2** **Khái niệm và cấu trúc mạng SDN** 3](#_Toc90647350)

[*1.2.1* *SDN là gì?* 3](#_Toc90647351)

[*1.2.2* *Cấu trúc mạng SDN* 3](#_Toc90647352)

[**1.3** **Giao thức OpenFlow** 5](#_Toc90647353)

[**1.4** **Cách triển khai của SDN** 6](#_Toc90647354)

[**1.5** **Ưu và nhược điểm của SDN** 6](#_Toc90647355)

[*1.5.1 Ưu điểm của SDN* 6](#_Toc90647356)

[*1.5.2 Nhược điểm của SDN* 8](#_Toc90647357)

[**1.6** **Ứng dụng của SDN** 9](#_Toc90647358)

[Chương 2: Nghiên cứu về controller ONOS 10](#_Toc90647359)

[**2.1 Giới thiệu ONOS** 10](#_Toc90647360)

[**2.2 Các thành phần của ONOS** 10](#_Toc90647361)

[*2.2.1 Dịch vụ và hệ thống con* 11](#_Toc90647362)

[*2.2.2 Cấu trúc hệ thống con* 12](#_Toc90647363)

[*2.2.3 Các nhà cung cấp* 12](#_Toc90647364)

[2.3 Sự giống và khác của ONOS với các controller khác 12](#_Toc90647365)

[Chương 3: Công cụ mô phỏng mininet 13](#_Toc90647366)

[**3.1 Giới thiệu Mininet** 13](#_Toc90647367)

[**3.2 Cài đặt Mininet** 13](#_Toc90647368)

[3.3 So sánh Mininet với các công cụ khác 13](#_Toc90647369)

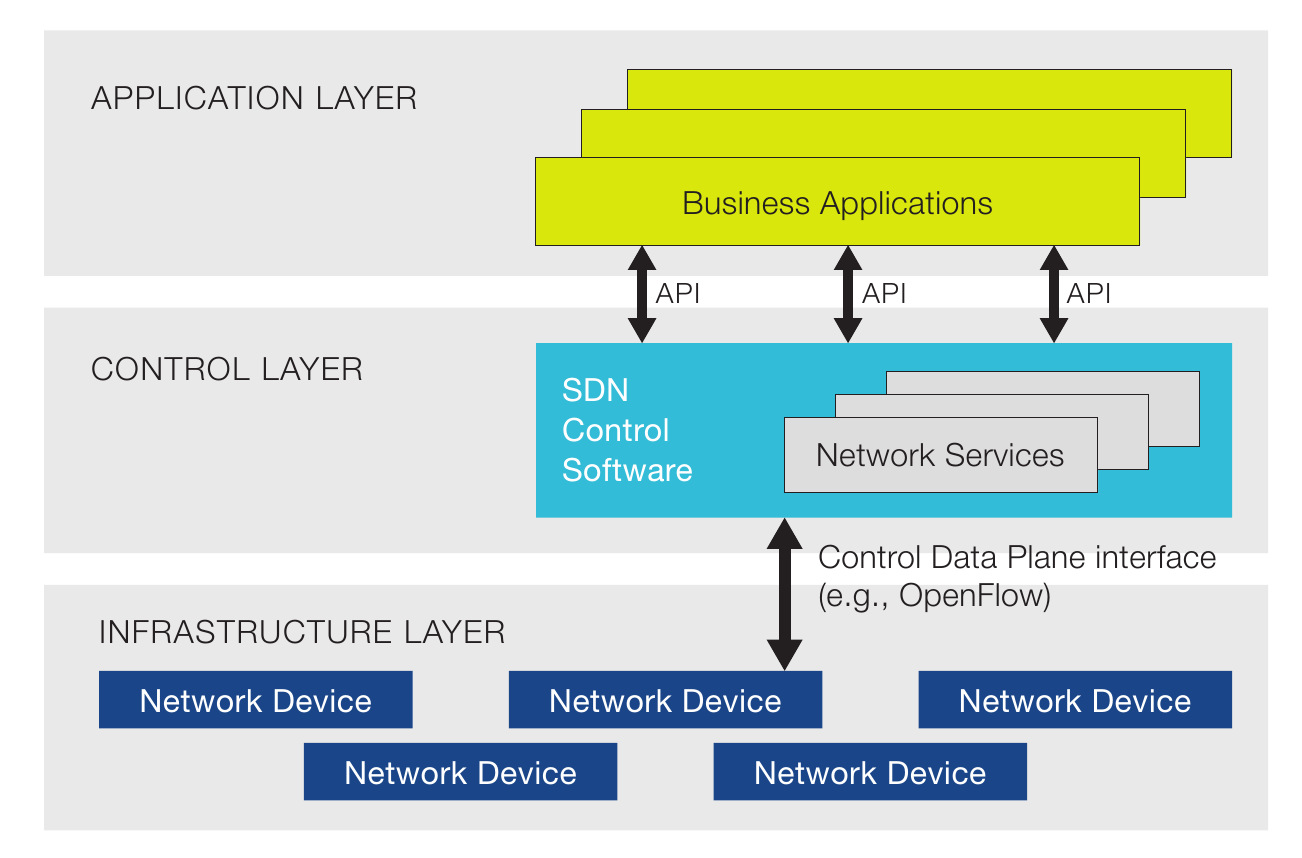
[*3.3.1 Lợi ích của mininet* 13](#_Toc90647370)

[*3.3.2 Hạn chế của Mininet* 14](#_Toc90647371)

# Chương 1: Tổng quan về SDN

* 1. **Giới thiệu chung**
* 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 xây dựng với các tầng của thiết bị chuyển mạch Ethernet, được sắp xếp theo c 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.
  1. **Khái niệm và cấu trúc mạng SDN**
     1. *SDN là gì?*
* SDN (Software-Defined Networking) là một kiến trúc có khả năng trừu tượng hóa các layer khác nhau của mạng, giúp mạng trở nên linh hoạt hơn. Mục tiêu chính của SDN là cải thiện khả năng kiểm soát mạng bằng cách cho phép các doanh nghiệp và nhà cung cấp dịch vụ đáp ứng nhanh chóng khi các nhu cầu kinh doanh thay đổi.
* Trong một mạng SDN, quản trị viên có thể định hình lưu lượng từ một console điều khiển tập trung mà không cần phải xử lý từng switch ở trong mạng. Bộ điều khiển SDN tập trung sẽ định hướng cho các switch cung cấp dịch vụ mạng ở bất cứ nơi nào ta cần, với mọi kết nối giữa server và thiết bị.
  + 1. *Cấu trúc mạng SDN*

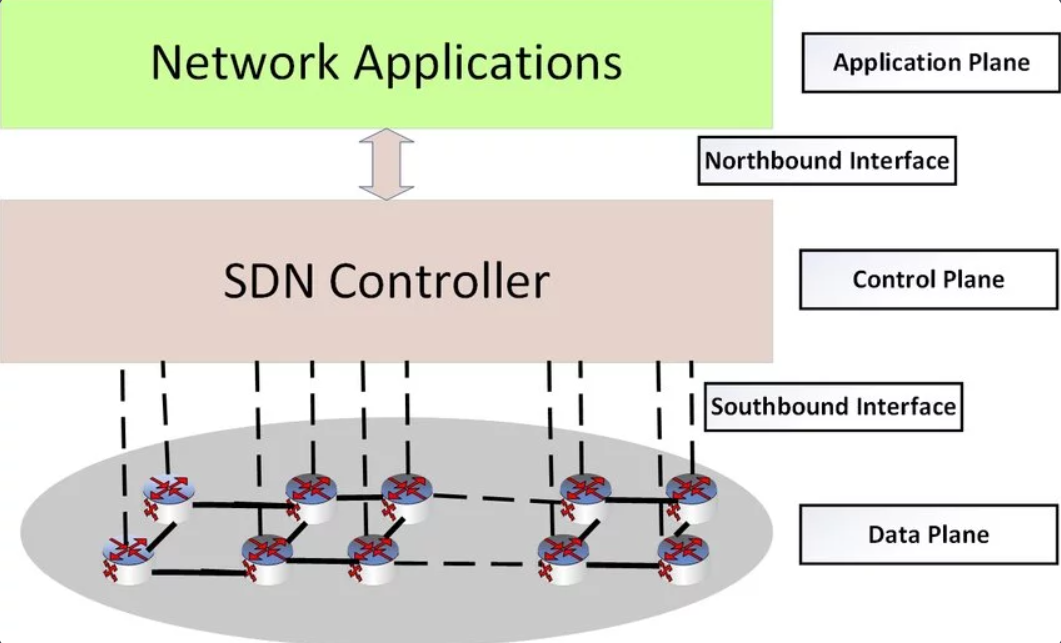
Về cơ bản, SDN được chia làm ba lớp: lớp ứng dụng (Application Layer), lớp điều khiển (Control Layer) và lớp thiết bị hạ tầng (Infrastructure Layer). Các lớp sẽ liên kết với nhau thông qua giao thức hoặc các API.



Hình 1. Cấu trúc mạng SDN 0‑1

* Lớp ứng dụng:
* Lớp ứng dụng (Application Layer): là các ứng dụng được triển khai trên mạng, 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 lập trình giúp cho hệ thống mạng để tối ưu hoạt động theo một yêu cầu nhất định.
* Lớp điều khiển:
* Lớp điều khiển (Control layer): là nơi tập trung các controller thực hiện việc điều khiển cấu hình mạng theo các yêu cầu từ lớp ứng dụng và khả năng của mạng. Các controller này có thể là các phần mềm được lập trình.
* Một Controller (bộ điều khiển) là một ứng dụng quản lý kiểm soát luồng lưu lượng trong môi trường mạng. Để điều khiển lớp cơ sở hạ tầng, lớp điều khiển sử dụng các cơ chế như OpenFlow, ONOS, ForCES, PCEP, NETCONF, SNMP hoặc thông qua các cơ chế riêng biệt. Hầu hết các SDN controller hiện nay dựa trên giao thức OpenFlow.
* SDN controller hoạt động như một loại hệ điều hành (OS) cho mạng. Tất cả thông tin liên lạc giữa các ứng dụng và các thiết bị phải đi qua controller. Controller sử dụng giao thức OpenFlow để cấu hình các thiết bị mạng và chọn đường đi tốt nhất cho các lưu lượng ứng dụng. 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.
* Vai trò:
* 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 thiết bị hạ tầng:
* Lớp thiết bị hạ tầng (Infrastructure Layer) của hệ thống mạng, bao gồm 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. Một thiết bị mạng có thể hoạt động theo sự điều khiển của nhiều controller khác nhau, điều này giúp tăng cường khả năng ảo hóa của mạng.
  1. **Giao thức OpenFlow**
* OpenFlow cho phép các bộ điều khiển mạng xác định đường đi của các gói mạng qua mạng chuyển mạch. Các bộ điều khiển khác biệt với các công tắc. Việc tách kiểm soát khỏi chuyển tiếp này cho phép quản lý lưu lượng phức tạp hơn là khả thi bằng cách sử dụng danh sách kiểm soát truy cập (ACL) và giao thức định tuyến. Ngoài ra, OpenFlow cho phép các thiết bị chuyển mạch từ các nhà cung cấp khác nhau - thường là mỗi nhà cung cấp có giao diện và ngôn ngữ kịch bản độc quyền của riêng họ - được quản lý từ xa bằng một giao thức mở duy nhất. Các nhà phát minh ra giao thức coi OpenFlow là một công cụ hỗ trợ mạng do phần mềm xác định (SDN).
* OpenFlow cho phép quản trị từ xa các bảng chuyển tiếp gói của công tắc lớp 3, bằng cách thêm, sửa đổi và loại bỏ các quy tắc và hành động đối sánh gói. Bằng cách này, các quyết định định tuyến có thể được thực hiện theo định kỳ hoặc đột xuấtbởi bộ điều khiển và được dịch thành các quy tắc và hành động có tuổi thọ có thể định cấu hình, sau đó được triển khai vào bảng luồng của bộ chuyển mạch, để chuyển tiếp thực tế các gói phù hợp tới bộ chuyển mạch ở tốc độ dây trong suốt thời gian của các quy tắc đó. Có thể chuyển tiếp các gói chưa được khớp bởi bộ chuyển mạch tới bộ điều khiển. Sau đó, bộ điều khiển có thể quyết định sửa đổi các quy tắc bảng luồng hiện có trên một hoặc nhiều bộ chuyển mạch hoặc triển khai các quy tắc mới, để ngăn chặn luồng lưu lượng có cấu trúc giữa bộ chuyển mạch và bộ điều khiển. Nó thậm chí có thể quyết định chuyển tiếp lưu lượng truy cập, miễn là nó đã yêu cầu bộ chuyển tiếp chuyển tiếp toàn bộ gói thay vì chỉ tiêu đề của chúng.
* Giao thức OpenFlow được xếp lớp trên Giao thức điều khiển truyền (TCP) và quy định việc sử dụng Bảo mật lớp truyền tải (TLS). Bộ điều khiển nên lắng nghe trên cổng TCP 6653 đối với các bộ chuyển mạch muốn thiết lập kết nối. Các phiên bản trước của giao thức OpenFlow sử dụng không chính thức cổng 6633. Một số triển khai mặt phẳng điều khiển mạng sử dụng giao thức để quản lý các phần tử chuyển tiếp mạng. OpenFlow chủ yếu được sử dụng giữa công tắc và bộ điều khiển trên một kênh an toàn.
  1. **Cách triển khai của SDN**
* Ban đầu, công nghệ Software-defined networking chỉ tập trung phân tách network control plane từ data plane. Trong đó control plan quyết định cách các packet sẽ truyền qua mạng, còn data plane thì di chuyển các packet từ nơi này sang nơi khác.
* Cách hoạt động điển hình là: một packet đến một switch mạng, và các quy tắc được tích hợp trong firmware cho biết nơi mà switch cần chuyển tiếp packet đó. Các quy tắc xử lý packet này được gửi đến switch từ một bộ điều khiển tập trung.
* Switch – hay còn gọi là thiết bị data plane – truy vấn controller để được hướng dẫn nếu cần, đồng thời cung cấp cho controller thông tin về lưu lượng mà nó xử lý. Switch sẽ gửi mọi packet đến cùng một điểm đến, theo cùng một đường dẫn và xử lý theo một cách duy nhất.
* SDN sử dụng một cơ chế hoạt động gọi là adaptive hay dynamic. Trong đó, một switch đưa ra một route request cho controller đối với một packet không có route cụ thể. Quá trình này tách biệt với adaptive routing – đưa ra các route request thông qua router với các thuật toán dựa trên cấu trúc liên kết mạng, chứ không phải thông qua một controller.
  1. **Ưu và nhược điểm của SDN**

*1.5.1 Ưu điểm của SDN*

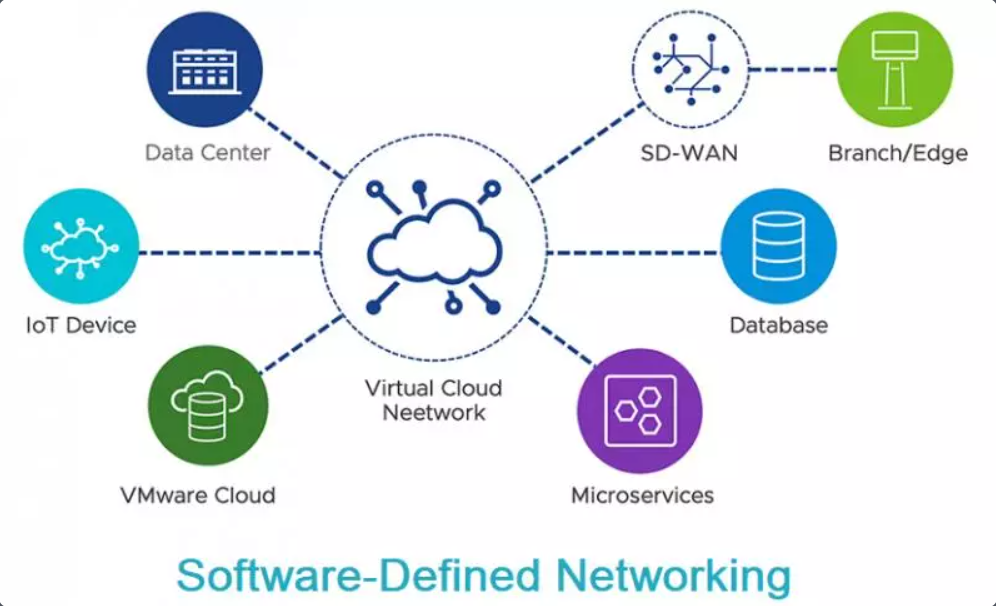


Hình 2. Ưu điểm của SDN 0‑1

SDN có rất nhiều ưu điểm khác nhau, chẳng hạn như:

* Với SDN, các quản trị viên có thể thay đổi bất kỳ quy tắc nào của switch khi cần – ưu tiên, loại bỏ hay thậm chí chặn các packet cụ thể với mức độ kiểm soát và bảo mật có độ chi tiết cao. Việc này đặc biệt hữu ích trong kiến trúc điện toán đám mây có nhiều người sử dụng, vì nó cho phép người quản trị quản lý lưu lượng một cách hiệu quả, linh hoạt. Về cơ bản, nó cho phép người quản trị sử dụng các switch có chi phí thấp hơn, có nhiều quyền kiểm soát hơn với các luồng lưu lượng mạng.
* Một số lợi ích khác nữa là quản lý mạng và hiển thị end-to-end. Người quản trị chỉ cần xử lý một controller tập trung để phân phối policy đến các switch được kết nối với nhau. Điều này trái ngược với việc cấu hình nhiều thiết bị riêng lẻ. Nó cũng là một lợi thế bảo mật, vì controller có thể giám sát lưu lượng và triển khai các chính sách bảo mật khác. Ví dụ: nếu controller cho rằng có lưu lượng truy cập đáng ngờ, nó có thể dễ dàng định tuyến lại, hoặc drop các packet đó.
* SDN còn có thể ảo hóa phần cứng và dịch vụ mà trước đây thường được xử lý bởi phần cứng chuyên dụng. Từ đó giúp giảm diện tích phần cứng và chi phí hoạt động.
* Bên cạnh đó, SDN cũng góp phần vào sự ra đời của công nghệ SD-WAN (Software-defined wide area network). SD-WAN sử dụng khả năng virtual overlay của công nghệ SDN. Nó trừu tượng hóa các liên kết kết nối của tổ chức thông qua một WAN, tạo ra một mạng ảo để có thể sử dụng bất kỳ kết nối nào mà controller cho là phù hợp để gửi lưu lượng.

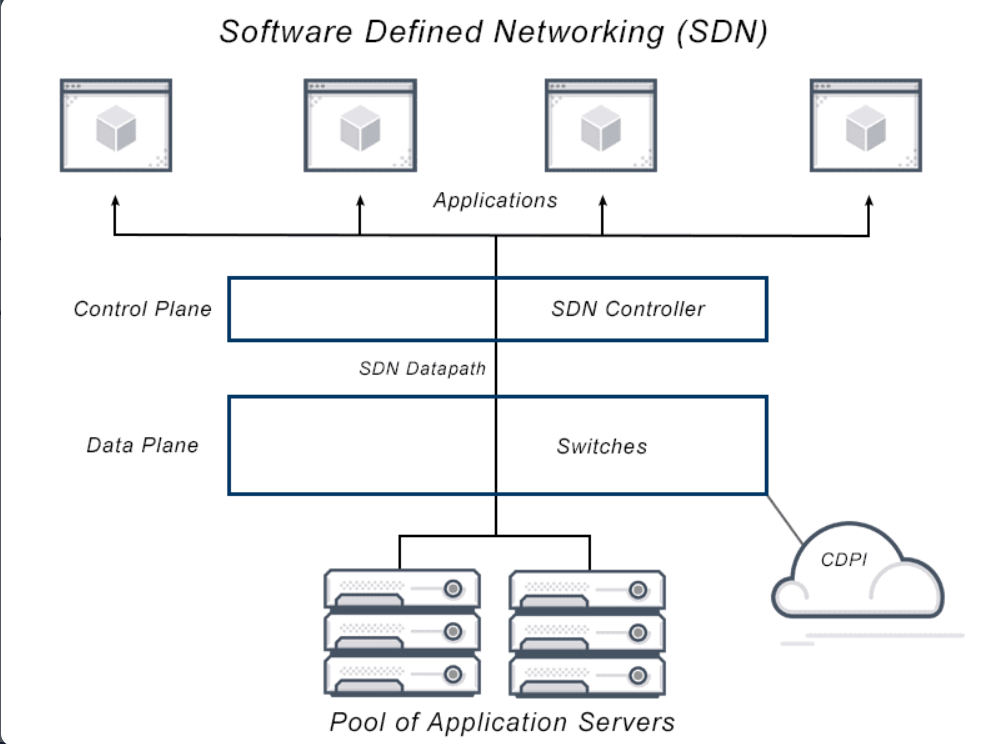
### *1.5.2 Nhược điểm của SDN*



Hình 3. Nhược điểm của SDN 0‑1

Đối tượng chính sử dụng SDN là những nhà cung cấp dịch vụ, nhà khai thác mạng, viễn thông, cùng một số công ty lớn như Facebook hay Google. Tất cả đều có nguồn lực to lớn để giải quyết và đóng góp vào một công nghệ mới nổi như SDN. Tuy nhiên, vẫn còn một số thách thức như:

* Khả năng bảo mật của SDN là một con dao hai lưỡi. Vì nếu bộ điều khiển tập trung có lỗi, kẻ tấn công có thể tấn công vào toàn mạng.
* Trớ thêu thay, một khó khăn khác nữa cho SDN là: trong ngành công nghiệp mạng không hề có định nghĩa về “Software-defined networking”. Các nhà cung cấp khác nhau cung cấp các cách tiếp cận khác nhau cho SDN, từ mô hình tập trung phần cứng và nền tảng ảo hóa cho đến các thiết kế mạng siêu hội tụ hay những phương pháp không cần controller.
* Một số sáng kiến mạng thường bị nhầm lần với SDN, gồm kết nối mạng white box, phân tách mạng, tự động hóa mạng hay mạng có khả năng lập trình. Mặc dù Software-defined networking có thể tận dụng ưu điểm của những công nghệ vừa kể đến, chúng vẫn là những công nghệ hoàn toàn khác biệt. SDN nổi lên vào năm 2011 khi nó được giới thiệu cùng với giao thức OpenFlow. Kể từ đó, việc ứng dụng Software-defined networking lại diễn ra tương đối chậm, đặc biệt là ở các doanh nghiệp có mạng nhỏ hoặc ít tài nguyên. Ngoài ra, nhiều doanh nghiệp cũng cho rằng chi phí triển khai SDN là một thách thức lớn.
  1. **Ứng dụng của SDN**

****

Hình 4. Ứng dụng của SDN 0‑1

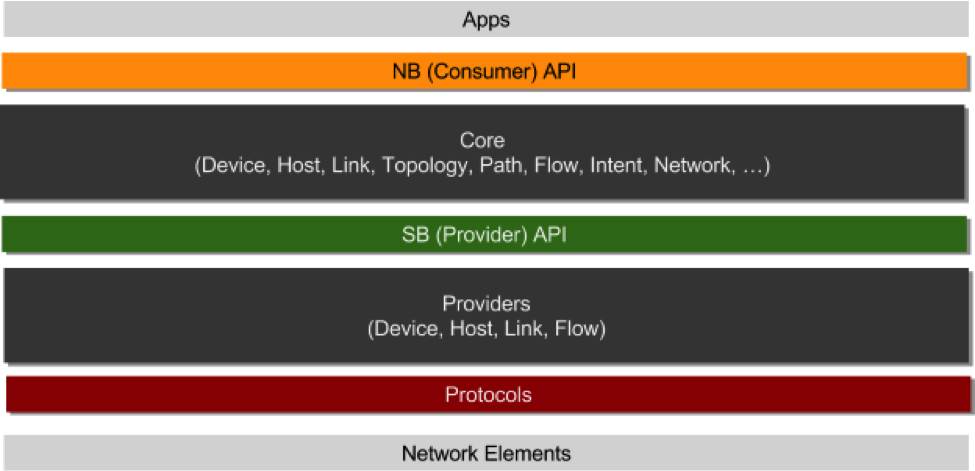
* DevOps – phương pháp tiếp cận dựa trên SDN có thể giúp tự động hóa các bản cập nhật và triển khai ứng dụng, từ đó tạo điều kiện thuận lợi cho DevOps. Cụ thể, SDN giúp tự động hóa các thành phần có sở hạ tầng khi các ứng dụng và nền tảng của DevOps được triển khai.
* Campus network (mạng campus) – Các 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.
* Bảo mật trung tâm dữ liệu – SDN hỗ trợ bảo vệ tập trung hơn và đơn giản hóa việc quản trị firewall. Nói chung, một doanh nghiệp phụ thuộc vào tương lửa để bảo mật trung tâm dữ liệu của nó. Tuy nhiên, một trung tâm có thể tạo ra một hệ thống tường lửa phân tán bằng cách thêm các tường lửa ảo để bảo vệ các máy ảo. Lớp bảo mật bổ sung này giúp ngăn vi phạm giữa các máy ảo. Ngoài ra, kiểm soát tập trung và tự động hóa SDN cho phép admin có thể xem, sửa đổi và kiểm soát hoạt động mạng để giảm nguy cơ vi phạm.

# Chương 2: Nghiên cứu về controller ONOS

## **2.1 Giới thiệu ONOS**

ONOS là viết tắt của **O**pen **N**etwork **O**perating **S**ystem. ONOS cung cấp mặt phẳng điều khiển cho mạng do phần mềm xác định (SDN), quản lý các thành phần mạng, chẳng hạn như thiết bị chuyển mạch và liên kết, đồng thời chạy các chương trình hoặc mô-đun phần mềm để cung cấp dịch vụ truyền thông tới các máy chủ cuối và các mạng lân cận.

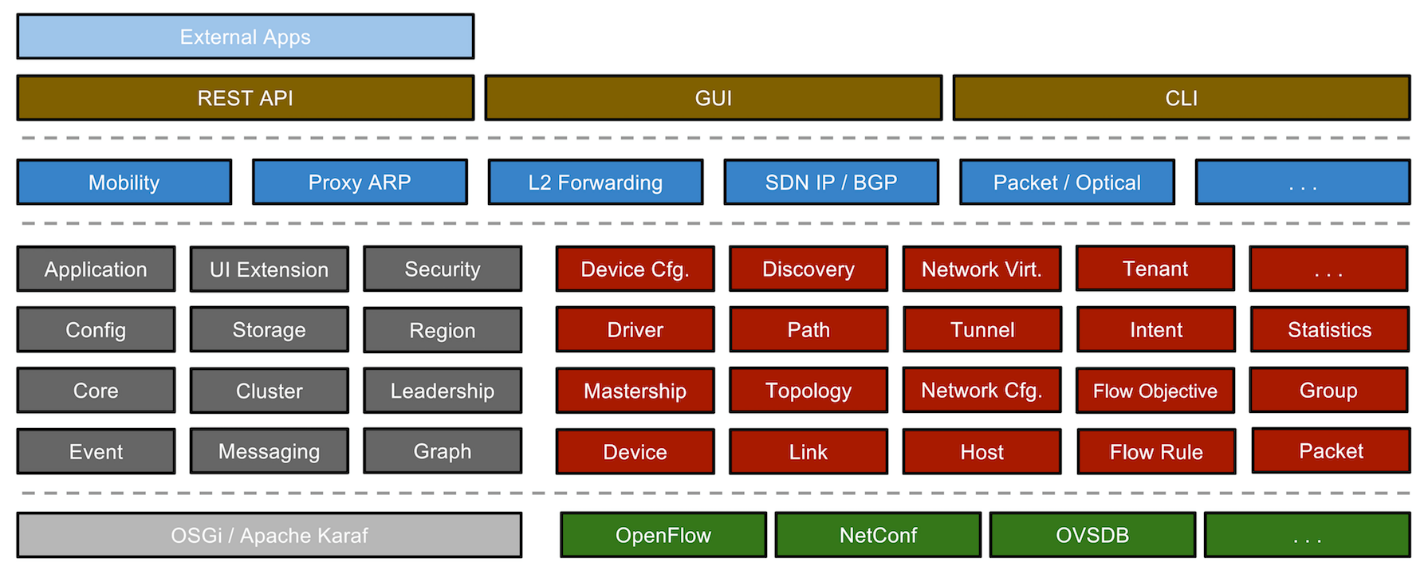
## **2.2 Các thành phần của ONOS**



Hình 5. Bậc hệ thống 0‑1

### *2.2.1 Dịch vụ và hệ thống con*

* Một ***dịch vụ***là một đơn vị chức năng mà bao gồm nhiều thành phần tạo ra một lát dọc qua các tầng như một chồng phần mềm. Chúng tôi đề cập đến tập hợp các thành phần tạo nên dịch vụ như một ***hệ thống con****.*Chúng tôi sử dụng các thuật ngữ 'dịch vụ' và 'hệ thống con' thay thế cho nhau trong hướng dẫn này.
* ONOS định nghĩa một số dịch vụ chính:
* *Hệ thống con thiết bị* - Quản lý kho thiết bị cơ sở hạ tầng.
* *Hệ thống con liên kết* - Quản lý khoảng không quảng cáo của các liên kết cơ sở hạ tầng.
* *Hệ thống con Máy chủ lưu trữ* - Quản lý khoảng không quảng cáo của các máy chủ trạm cuối và vị trí của chúng trên mạng.
* *Hệ thống con cấu trúc liên kết* - Quản lý ảnh chụp nhanh theo thứ tự thời gian của các chế độ xem biểu đồ mạng.
* *PathService* - Tính toán / tìm đường dẫn giữa các thiết bị cơ sở hạ tầng hoặc giữa các máy chủ trạm đầu cuối bằng cách sử dụng ảnh chụp nhanh biểu đồ cấu trúc liên kết gần đây nhất.
* *Hệ thống con FlowRule* - Quản lý khoảng không quảng cáo của các quy tắc luồng đối sánh / hành động được cài đặt trên các thiết bị cơ sở hạ tầng và cung cấp các chỉ số luồng.
* *Hệ thống con gói* - Cho phép ứng dụng lắng nghe các gói dữ liệu nhận được từ các thiết bị mạng và phát các gói dữ liệu ra mạng thông qua một hoặc nhiều thiết bị mạng.
* Hình sau minh họa các hệ thống con khác nhau là một phần của ONOS ngày nay và một số hệ thống con được lên kế hoạch trong tương lai gần:

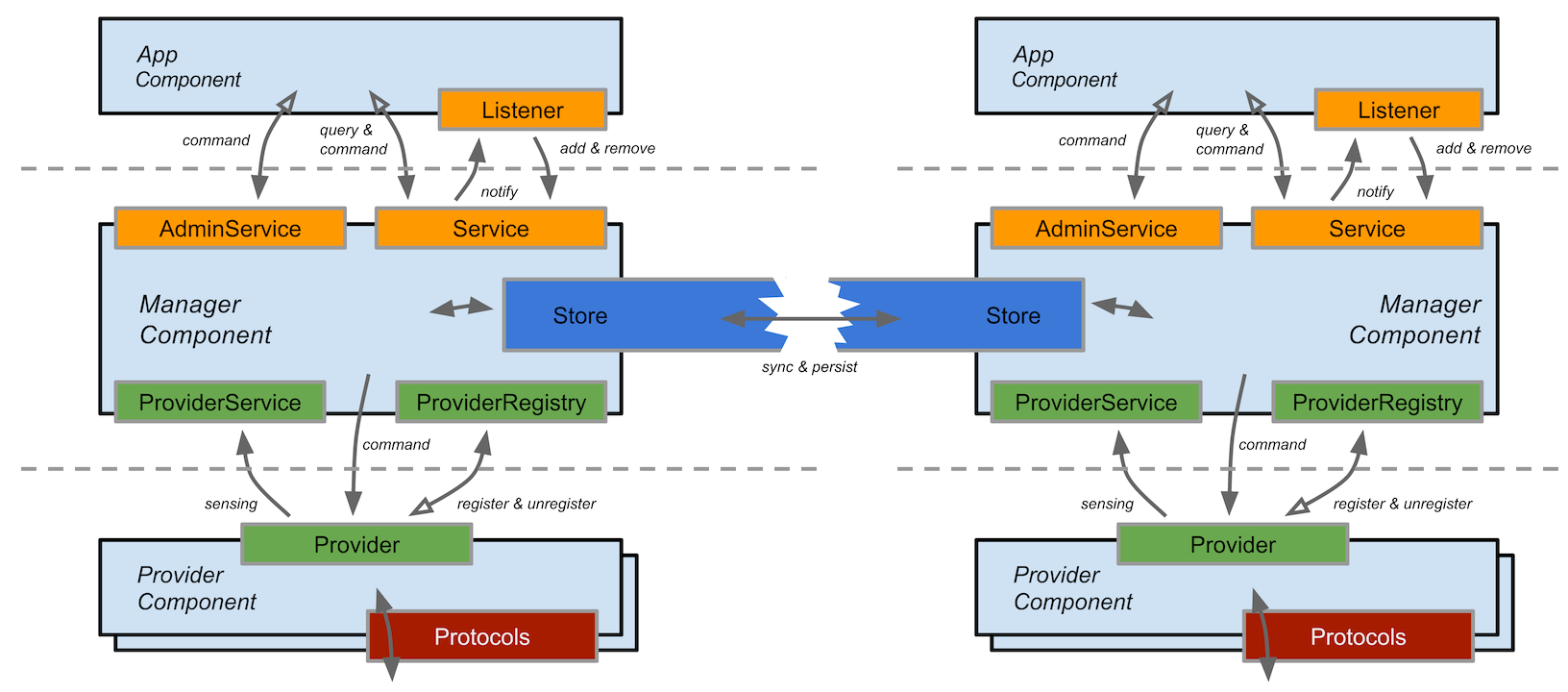


Hình 6. Các hệ thống con 0‑1

### *2.2.2 Cấu trúc hệ thống con*

Mỗi thành phần của hệ thống con nằm ở một trong ba lớp chính và có thể được xác định bằng một hoặc nhiều Giao diện Java mà chúng thực thi.

Hình dưới đây tóm tắt các mối quan hệ của các thành phần hệ thống con. Các đường chấm trên cùng và dưới cùng trong hình biểu thị ranh giới giữa các tầng được tạo bởi API hướng bắc và hướng nam tương ứng.



Hình 7. Cấu trúc hệ thống con 0‑1

### *2.2.3 Các nhà cung cấp*

Tầng thấp nhất của ngăn xếp ONOS, Nhà cung cấp giao diện với mạng thông qua các thư viện theo giao thức cụ thể và với lõi thông qua ProviderServicegiao diện.

Các nhà cung cấp nhận thức về giao thức có trách nhiệm tương tác với môi trường mạng bằng cách sử dụng các giao thức điều khiển và cấu hình khác nhau, đồng thời cung cấp dữ liệu cảm nhận về dịch vụ cụ thể cho lõi. Nhà cung cấp cũng có thể thu thập dữ liệu từ các hệ thống con khác, chuyển đổi chúng thành dữ liệu dành riêng cho dịch vụ.

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

## 2.3 Sự giống và khác của ONOS với các controller khác

* ONOS cung cấp một số loại chức năng tương tự, bao gồm API và phần tóm tắt, phân bổ tài nguyên và quyền, cũng như phần mềm hướng tới người dùng như [CLI](https://wiki.onosproject.org/display/ONOS/The+ONOS+CLI) , [GUI](https://wiki.onosproject.org/display/ONOS/The+ONOS+Web+GUI) và các ứng dụng hệ thống .
* ONOS quản lý toàn bộ mạng của bạn thay vì một thiết bị duy nhất, điều này có thể đơn giản hóa đáng kể việc quản lý, cấu hình và triển khai phần mềm, phần cứng và dịch vụ mới.
* ONOS và các ứng dụng hoạt động như một bộ điều khiển SDN phân tán, có thể mở rộng.

# Chương 3: Công cụ mô phỏng mininet

## **3.1 Giới thiệu Mininet**

Mininet là một trình giả lập mạng, hay có lẽ chính xác hơn là một hệ thống điều phối mô phỏng mạng. Nó chạy một tập hợp các máy chủ cuối, bộ chuyển mạch, bộ định tuyến và liên kết trên một nhân Linux duy nhất. Nó sử dụng ảo hóa nhẹ để làm cho một hệ thống giống như một mạng hoàn chỉnh, chạy cùng một nhân, hệ thống và mã người dùng. Máy chủ Mininet hoạt động giống như một máy thực; bạn có thể ssh vào nó (nếu bạn bắt đầu sshd và kết nối mạng với máy chủ của bạn) và chạy các chương trình tùy ý (bao gồm bất kỳ thứ gì được cài đặt trên hệ thống Linux cơ bản.) Các chương trình bạn chạy có thể gửi các gói thông qua giao diện giống như giao diện Ethernet thực, với tốc độ và độ trễ liên kết nhất định. Các gói được xử lý bởi những gì trông giống như một bộ chuyển mạch Ethernet, bộ định tuyến hoặc hộp trung gian thực, với một số lượng hàng đợi nhất định. Khi hai chương trình, như iperf máy khách và máy chủ, giao tiếp thông qua Mininet, hiệu suất đo được phải khớp với hiệu suất của hai máy gốc (chậm hơn).

## **3.2 Cài đặt Mininet**

## 3.3 So sánh Mininet với các công cụ khác

### *3.3.1 Lợi ích của mininet*

* Nó nhanh chóng - khởi động một mạng đơn giản chỉ mất vài giây. Điều này có nghĩa là vòng lặp chạy-chỉnh sửa-gỡ lỗi của bạn có thể rất nhanh chóng.
* Bạn có thể tạo cấu trúc liên kết tùy chỉnh: một công tắc duy nhất, cấu trúc liên kết giống Internet lớn hơn, đường trục Stanford, trung tâm dữ liệu hoặc bất kỳ thứ gì khác.
* Bạn có thể chạy các chương trình thực: bất cứ thứ gì chạy trên Linux đều có sẵn để bạn chạy, từ máy chủ web đến các công cụ giám sát cửa sổ TCP đến Wireshark.
* Bạn có thể tùy chỉnh chuyển tiếp gói: Bộ chuyển mạch của Mininet có thể lập trình được bằng giao thức OpenFlow. Có thể dễ dàng chuyển các thiết kế mạng do phần mềm xác định tùy chỉnh chạy trong Mininet sang bộ chuyển mạch OpenFlow phần cứng để chuyển tiếp gói tốc độ đường truyền.
* Bạn có thể chạy Mininet trên máy tính xách tay của mình, trên máy chủ, trong máy ảo, trên hộp Linux gốc (Mininet được bao gồm trong Ubuntu 12.10+!) Hoặc trên đám mây (ví dụ: Amazon EC2.)
* Bạn có thể chia sẻ và sao chép kết quả: bất kỳ ai có máy tính đều có thể chạy mã của bạn sau khi bạn đã đóng gói nó.
* Bạn có thể sử dụng nó một cách dễ dàng: bạn có thể tạo và chạy các thử nghiệm Mininet bằng cách viết các tập lệnh Python đơn giản (hoặc phức tạp nếu cần).
* Mininet là một dự án mã nguồn mở, vì vậy bạn được khuyến khích kiểm tra mã nguồn của nó trên https://github.com/mininet, sửa đổi nó, sửa lỗi, sự cố tệp / yêu cầu tính năng và gửi các bản vá / yêu cầu kéo. Bạn cũng có thể chỉnh sửa tài liệu này để sửa bất kỳ lỗi nào hoặc thêm giải thích hoặc thông tin bổ sung.
* Mininet đang được phát triển tích cực. Vì vậy, nếu nó tệ, không có ý nghĩa hoặc không hoạt động vì lý do nào đó, vui lòng cho chúng tôi biết trên mininet-discuss và cộng đồng người dùng và nhà phát triển Mininet có thể cố gắng giải thích, sửa lỗi hoặc giúp bạn khắc phục. :-) Nếu bạn tìm thấy lỗi, bạn nên gửi các bản vá để sửa chúng hoặc ít nhất là gửi một vấn đề trên github bao gồm một trường hợp thử nghiệm có thể tái tạo.

### *3.3.2 Hạn chế của Mininet*

* Chạy trên một hệ thống đơn lẻ rất tiện lợi, nhưng nó áp đặt giới hạn tài nguyên: nếu máy chủ của bạn có CPU 3 GHz và có thể chuyển khoảng 10 Gbps lưu lượng mô phỏng, thì những tài nguyên đó sẽ cần được cân bằng và chia sẻ giữa các máy chủ ảo và thiết bị chuyển mạch 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 nhân hệ điều hành khác. (Mặc dù bạn có thể gắn máy ảo vào Mininet.)
* Mininet sẽ không viết bộ điều khiển OpenFlow cho bạn; nếu bạn cần hành vi chuyển đổi hoặc định tuyến tùy chỉnh, bạn sẽ cần tìm hoặc phát triển một bộ điều khiển với các tính năng bạn yêu cầu.
* Theo mặc định, mạng Mininet của bạn được cách ly khỏi mạng LAN và với internet - đây thường là một điều tốt! Tuy nhiên, bạn có thể sử dụng NATđối tượng và / hoặc --nattùy chọn để kết nối mạng Mininet 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 examples/hwintf.py để biết chi tiết.)
* 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 các daemon yêu cầu cấu hình trong / etc, và bạn cần phải cẩn thận để không giết nhầm các quy trình do nhầm lẫn. (Lưu ý rằng bind.py ví dụ minh họa cách có các thư mục riêng trên mỗi máy chủ lưu trữ.)
* Không giống như một trình mô phỏng, Mininet không có khái niệm mạnh mẽ về thời gian ảo; điều này có nghĩa là các phép đo thời gian sẽ dựa trên thời gian thực và kết quả nhanh hơn thời gian thực (ví dụ: mạng 100 Gbps) không thể dễ dàng được mô phỏng.