TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO CUỐI KÌ MÔN KIẾN TRÚC INTERNET

THÁCH THỨC TRONG HIỆN THỰC SOFTWARE DEFINED NETWORKING

Người hướng dẫn: **TS TRẦN TRUNG TÍN**
Người thực hiện: **VÕ THỊ LAN CHI - 52200320**
Lớp **: 22050401**
Khoá **: 26**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2025

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO CUỐI KÌ MÔN KIẾN TRÚC INTERNET

THÁCH THỨC TRONG HIỆN THỰC SOFTWARE DEFINED NETWORKING

Người hướng dẫn: **TS TRẦN TRUNG TÍN**
Người thực hiện: **VÕ THỊ LAN CHI - 52200320**
Lớp **: 22050401**
Khoá **: 26**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2025

LỜI CẢM ƠN

Em xin gửi lời cảm ơn sâu sắc đến thầy *Trần Trung Tín*. Trong quá trình tìm hiểu và học tập môn *Kiến trúc Internet*, em đã nhận được sự giảng dạy và hướng dẫn rất tận tình, tâm huyết của thầy. Thầy đã giúp em tích lũy thêm nhiều kiến thức hay và bổ ích. Từ những kiến thức mà thầy truyền đạt, em xin trình bày lại những gì mình đã tìm hiểu về chủ đề cuối kì gửi đến thầy.

Tuy nhiên em tự nhận thấy bản thân còn hạn chế nhiều về môn *Kiến trúc Internet* nên không thể tránh khỏi những thiếu sót trong quá trình hoàn thành bài báo cáo quá trình này. Mong thầy xem và góp ý để bài báo cáo của em được cải thiện hơn.

Em xin chân thành cảm ơn thầy vì đã hỗ trợ em trong quá trình thực hiện bài báo cáo này!

ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi và được sự hướng dẫn của TS Trần Trung Tín. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày tháng năm

Tác giả

(ký tên và ghi rõ họ tên)

CHI

Võ Thị Lan Chi

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

_____

_____

_____

_____

_____

_____

_____

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

_____

_____

_____

_____

_____

_____

_____

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

TÓM TẮT

Bài báo cáo này trình bày đề tài về những thách thức trong hiện thực của SDN, mạng xác định bằng phần mềm, một mô hình mạng mới trong ngành công nghệ thông tin nói chung và mạng máy tính nói riêng. Báo cáo được chia thành 4 chương:

- Chương 1: Chương này trình bày bối cảnh và lý do chọn đề tài nghiên cứu về SDN trong xu thế phát triển mạng hiện nay. Bên cạnh đó, cũng nêu rõ mục tiêu nghiên cứu và phương pháp nghiên cứu được sử dụng trong quá trình thực hiện đề tài.
- Chương 2: Chương này cung cấp các kiến thức cơ bản về SDN, bao gồm định nghĩa, đặc điểm và lợi ích khi so sánh với mô hình mạng truyền thống. Ngoài ra, cũng trình bày kiến trúc tổng thể của SDN và cách thức hoạt động giữa các thành phần trong SDN.
- Chương 3: Thảo luận về những thách thức trong hiện thực của SDN khi áp dụng vào môi trường doanh nghiệp, trung tâm dữ liệu, ISP và điện toán đám mây. Bên cạnh đó là những biện pháp được đề cập sơ lược cho từng thách thức.
- Chương 4: Chương cuối cùng tổng kết lại nội dung chính của bài báo cáo, đánh giá những kết quả đạt được, đồng thời nêu ra những thuận lợi và khó khăn trong quá trình nghiên cứu. Qua đó, đưa ra định hướng cho các nghiên cứu tiếp theo liên quan đến việc hiện thực và phát triển SDN.

Báo cáo đã trình bày tổng quan về SDN và phân tích các thách thức khi triển khai trong thực tế. Một số giải pháp được đề xuất nhằm cải thiện hiệu quả và bảo mật, góp phần hỗ trợ việc ứng dụng SDN trong các môi trường mạng hiện đại.

MỤC LỤC

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

CÁC CHỮ VIẾT TẮT

Tiếng Việt

CNTT	Công nghệ thông tin

Tiếng Anh

SDN	Software-Defined Networking
ONF	One Networking Foundation
WAN	Wide Area Network
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
API	Application Programming Interface
IoT	Internet of Things
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
BGP	Border Gateway Protocol
ISP	Internet Service Provider
NOX	NOX OpenFlow Controller
NOX-MT	NOX Multi-Threaded
RTT	Round Trip Time
MPLS	Multi-Protocol Label Switching
VM	Virtual Machine
VLAN	Virtual Local Area Network
VXLAN	Virtual Extensible Local Area Network
ARP	Address Resolution Protocol
TLS/DTLS	Transport Layer Security/Datagram Transport Layer Security
OSPF	Open Shortest Path First
QoS	Quality of Service

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

Bảng biểu

Hình vẽ

CHƯƠNG 1 - MỞ ĐẦU

1.1 Lý do chọn đề tài

Sự phát triển của mạng Internet đã tạo nên một cuộc cách mạng trong lĩnh vực công nghệ thông tin, đem lại nhiều lợi ích cho con người như giao tiếp dễ dàng, trao đổi kiến thức và đặt nền móng cho nền kinh tế tri thức ngày nay. Tuy nhiên, với sự tiến bộ không ngừng của công nghệ, các mô hình mạng truyền thống không còn đáp ứng được những nhu cầu ngày càng cao của các doanh nghiệp, tổ chức và người dùng. Các mô hình mạng ngày nay cần có khả năng thay đổi nhanh chóng về các thông số như tốc độ, độ trễ, băng thông, định tuyến và bảo mật để đáp ứng yêu cầu đa dạng của những ứng dụng hiện đại.

Trong bối cảnh đó, SDN ra đời như một giải pháp tiềm năng, với khả năng trừu tượng hoá các lớp mạng, tách biệt lớp điều khiển và lớp dữ liệu để giúp quản lý mô hình mạng linh hoạt và hiệu quả hơn. Tuy nhiên, quá trình triển khai SDN trong thực tế vẫn đối mặt với nhiều thách thức lớn như hiệu suất, bảo mật và khả năng mở rộng. Do đó, việc nghiên cứu chi tiết các khó khăn này và đề xuất giải pháp là tiền đề để thức đẩy việc ứng dụng SDN trong thực tế, đáp ứng nhu cầu ngày càng tăng của các hệ thống mạng hiện đại.

1.2 Mục tiêu nghiên cứu

Trong bối cảnh hạ tầng mạng truyền thống ngày càng thiếu linh hoạt và khó kiểm soát, SDN đã ra đời như là một giải pháp tiềm năng để cải thiện những thiếu sót của mạng truyền thống. Tuy nhiên, việc chuyển đổi mô hình đem lại hàng loạt vấn đề về kỹ thuật, vận hành và tổ chức. Vì vậy mà mục tiêu của bài báo cáo này là phân tích và làm rõ các thách thức trong quá trình hiện thực hoá mô hình mạng SDN trong môi trường thực tế. Cụ thể, mục tiêu nghiên cứu gồm:

- Tìm hiểu tổng quan kiến trúc và nguyên lý hoạt động của SDN: làm rõ cách thức mà SDN phân tách lớp điều khiển và lớp dữ liệu, cũng như là vai trò của thành phần như bộ điều khiển, switch SDN và giao thức liên kết OpenFlow.

- Phân tích các thách thức kỹ thuật khi triển khai SDN trong môi trường thực tế, bao gồm: độ trễ và hiệu năng, khả năng mở rộng của hệ thống, vấn đề tương thích giữa SDN và mạng truyền thống, cuối cùng là về bảo mật của SDN.
- Đánh giá các yếu tố phi kỹ thuật ảnh hưởng đến việc hiện thực SDN.
- Tổng hợp các thách thức và đề xuất những giải pháp nhằm khắc phục những khó khăn đó, từ đó đề xuất hướng đi tiềm năng để cải thiện hiệu quả triển khai SDN trong tương lai.

Thông qua các mục tiêu đã đề cập, bài báo cáo kỳ vọng sẽ đóng góp một góc nhìn hệ thống và thực tiễn về hiện thực triển khai SDN.

1.3 Phương pháp nghiên cứu

Để đạt được những mục tiêu đã đề ra, nghiên cứu sẽ kết hợp với phân tích tài liệu chuyên sâu nhằm thu thập, tổng hợp và đánh giá các thách thức trong quá trình hiện thức hoá mô hình SDN. Các phương pháp cụ thể bao gồm:

- Nghiên cứu tài liệu: Là phương pháp chủ đạo, nhằm thu thập và phân tích các công trình nghiên cứu trước đó liên quan đến SDN, bao gồm bài báo khoa học, luận văn, báo cáo kỹ thuật, cũng như các nghiên cứu ứng dụng từ các tổ chức công nghệ lớn. Đây là cơ sở lý thuyết để xây dựng nền tảng kiến thức vững chắc về kiến trúc SDN, các mô hình triển khai và các vấn đề nổi bật thường gặp.
- Phân tích so sánh: Phương pháp này tiến hành đối chiếu và so sánh giữa mô hình triển khai SDN với mô hình truyền thống, giữa các mô hình SDN khác nhau, nhằm làm rõ sự khác biệt về yêu cầu kỹ thuật, mục tiêu triển khai và các rào cản cụ thể. Ngoài ra, phương pháp này cũng giúp đánh giá hiệu quả của các giải pháp khắc phụ đã được đề xuất.
- Phân tích trường hợp điển hình: Một số tổ chức khi triển khai SDN trong các hệ thống mạng thực tế cũng gặp những trục trặc gây ảnh hưởng lớn. Những khó khăn này sẽ được phân tích để làm rõ các thách thức kỹ thuật và tổ chức

khi chuyển đổi từ mạng truyền thống sang SDN. Mọi ví dụ đều giúp kiểm chứng tính thực tiễn của các vấn đề lý thuyết đã được đề cập.

- Tổng hợp và đánh giá: Sau khi thu thập và phân tích, bài báo cáo sẽ tiến hành tổng hợp các nhóm thách thức chính, phân loại theo từng khía cạnh, đồng thời đánh giá mức độ ảnh hưởng và khả năng ứng dụng của các giải pháp hiện có. Từ đó, đề xuất một số định hướng nghiên cứu và triển khai tiếp theo.

Thông qua việc kết hợp các phương pháp trên nhằm đưa ra cái nhìn toàn diện và có hệ thống về các thách thức trong hiện thực hoá SDN. Ngoài ra, việc tổng hợp và đánh giá giúp xác định rõ nguyên nhân cốt lõi của các vấn đề đã và đang tồn tại, từ đó làm cơ sở để đề xuất những định hướng cải tiến và giải pháp khả thi giúp hỗ trợ việc triển khai SDN hiệu quả trong thực tế.

CHƯƠNG 2 - TỔNG QUAN VỀ SDN

2.1 Giới thiệu SDN

2.1.1 Định nghĩa SDN

Khi bắt đầu làm quen với các từ thông dụng trong ngành CNTT như điện toán đám mây, IoT hoặc ảo hoá mạng, thì thế giới lại xuất hiện thêm một thuật ngữ mới: Mạng định nghĩa bằng phần mềm (Software-Defined Networking – SDN), và thu hút được sự quan tâm rất lớn từ giới học thuật và công nghiệp, nhanh chóng trở thành một chủ đề nóng mà mọi người chọn để nghiên cứu.

Trên thế giới có nhiều định nghĩa về SDN nhưng theo tổ chức ONF – 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 SDN được định nghĩa như sau: "Mạng định nghĩa bằng phần mềm hay Software-Defined Networking (SDN) là một kiểu kiến trúc mạng mới, năng 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 hay Data Plane), điều này cho phép việc điều khiển mạng có thể lập trình được dễ dàng và cơ sở hạ tầng mạng độc lập với các ứng dụng và dịch vụ mạng".

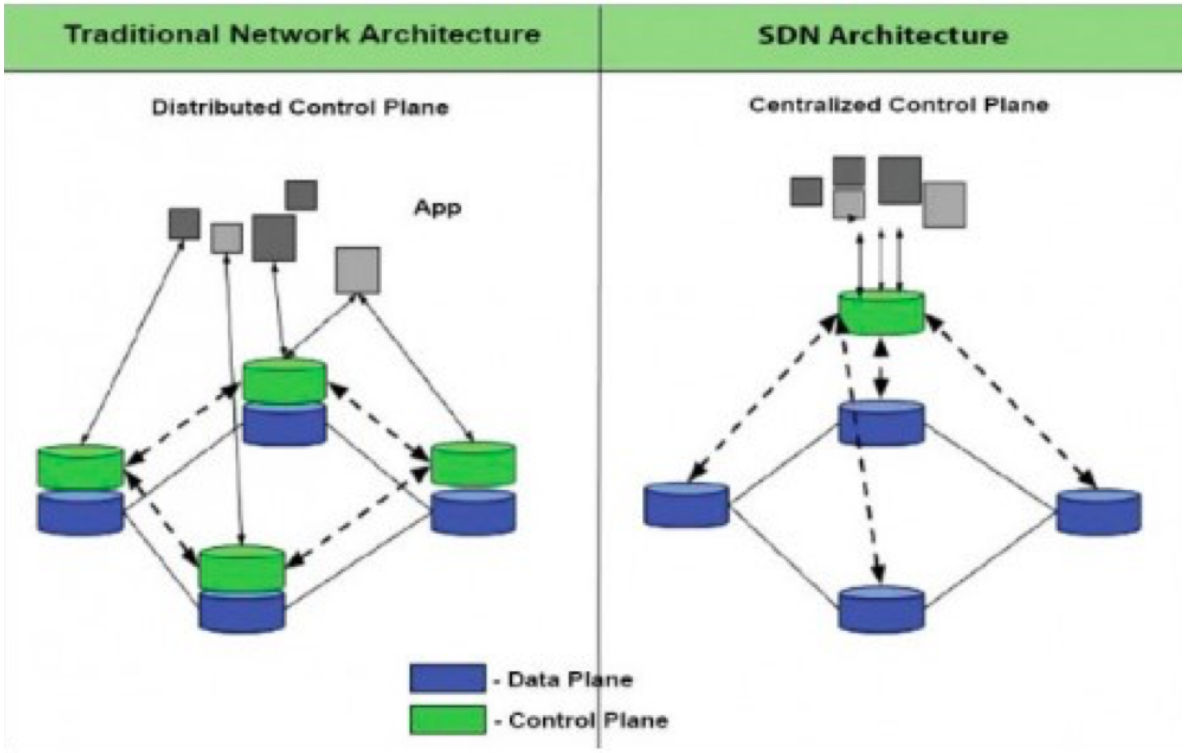
Theo như định nghĩa từ ONF, có thể hiểu rằng SDN là một kiến trúc có khả năng trừu tượng hoá các lớp khác nhau của mạng, giúp mạng trở nên linh hoạt hơn. Mục tiêu chính của SDN là cải thiện khả năng kiểm soát mạng bằng cách cho phép các doanh nghiệp và các nhà cung cấp dịch vụ đáp ứng nhanh chóng khi các nhu cầu kinh doanh thay đổi.

2.1.2 So sánh SDN với mạng truyền thống

Công nghệ SDN là một cách tiếp cận hiện đại, lấy cảm hứng từ công nghệ điện toán đám mây, cho phép mô hình có thể tập trung hoá việc điều khiển các thiết bị mạng. Thay vì phải cấu hình thủ công trên từng thiết bị riêng lẻ, gây tốn thời gian và công sức,

thì SDN sẽ giúp đơn giản hoá quá trình quản trị mạng bằng cách sử dụng các chương trình phần mềm có thể lập trình để cài đặt và quản lý toàn bộ hệ thống một cách linh hoạt và hiệu quả. Mô hình này còn hỗ trợ giám sát mạng tốt hơn và dễ dàng điều chỉnh khi cần thiết.

Không giống như SDN thì mạng truyền thống có hai đặc điểm nổi bật. Thứ nhất, hoạt động của mạng truyền thống chủ yếu dựa vào việc triển khai các thiết bị phần cứng chuyên dụng, như switch, router, firewall hoặc controller. Thứ hai, các chức năng mạng trong mô hình truyền thống đều được tích hợp trực tiếp vào phần cứng chuyên biệt.



Hình 2.1: So sánh mô hình SDN và mô hình truyền thống

Theo như hình minh hoạ, có thể thấy những sự khác biệt rõ rệt giữa SDN và mạng truyền thống:

Truyền thống	SDN
Phần điều khiển và phần dữ liệu được tích hợp trong thiết bị mạng	Phần điều khiển được tách riêng khỏi thiết bị mạng và được chuyển đến một thiết bị được gọi là bộ điều khiển
Việc thu thập và xử lý thông tin: Được thực hiện ở tất cả các phần tử trong mạng	Được tập trung xử lý ở bộ điều khiển
Không được lập trình bởi các ứng dụng, các thiết bị mạng phải được cấu hình một cách riêng lẻ và thủ công	Có thể lập trình bởi các ứng dụng, bộ điều khiển SDN có thể tương tác đến các thiết bị trong mạng
Do phần điều khiển tích hợp vào các thiết bị mạng, gây bất tiện trong khả năng giao tiếp với các thiết bị. Quản trị viên không thể dễ dàng truy cập để tuỳ chỉnh việc vận hành lưu lượng mạng khi cần.	Khả năng giao tiếp với các thiết bị tốt hơn. Cho phép tài nguyên được cung cấp từ một nguồn tập trung. Từ đó cung cấp quyền hạn cho quản trị viên có thể điều khiển lưu lượng mạng tại giao diện người dùng.

Bảng 2.1: Bảng so sánh mạng SDN và mạng truyền thống

Việc tách phần điều khiển tạo thành bộ điều khiển trung tâm, giúp cho các thiết bị mạng ở lớp thiết bị không cần phải hiểu và xử lý những thông tin hay giao thức phức tạp nào, mà chúng chỉ cần tiếp nhận và vận chuyển dữ liệu theo sự chỉ đạo từ bộ điều khiển SDN. (Nguyễn Thị Thảo et al., 2022)

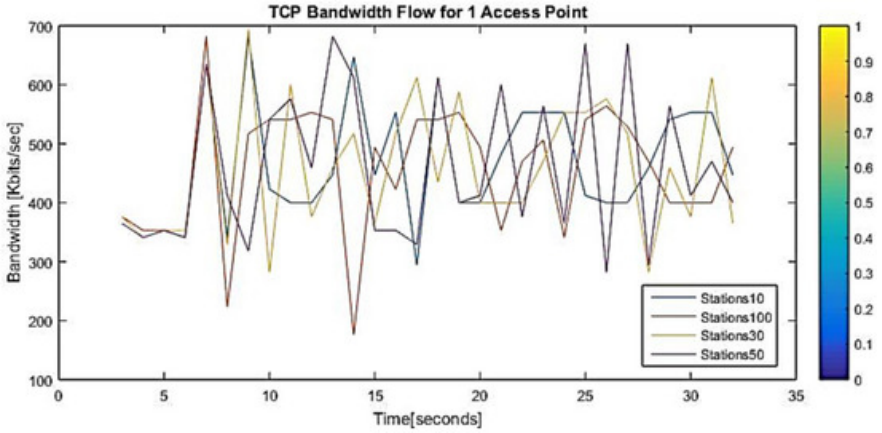
2.1.3 Lợi ích của SDN

Qua phần phân tích so sánh giữa SDN và mạng truyền thống, có thể thấy mô hình SDN đã giải quyết được những vấn đề mà mạng truyền thốngkhông thể đáp ứng được nhu cầu hiện đại. Bên cạnh đó, SDN cũng đem lại nhiều lợi ích cho công nghệ thông tin ngày nay, bao gồm:

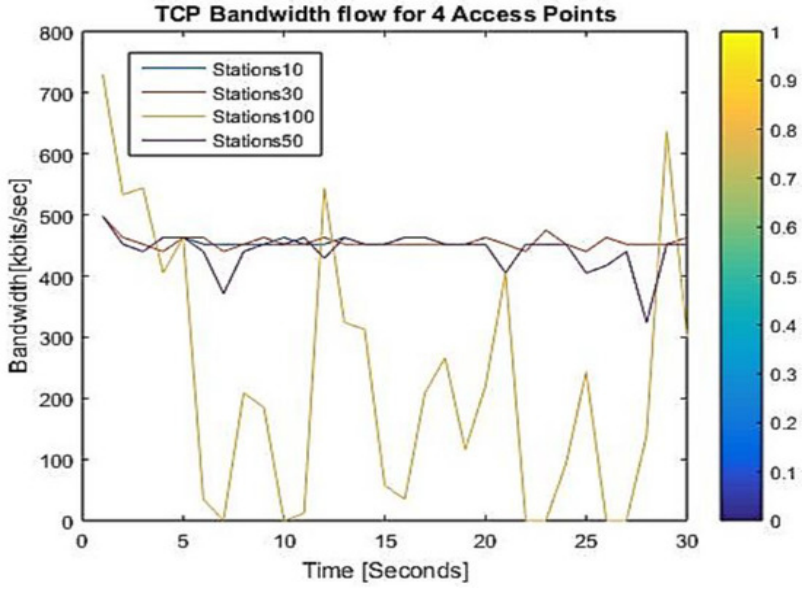
- **Quản lý mạng linh hoạt và dễ dàng hơn**: SDN cho phép quản lý mạng tập trung thông qua một bộ điều khiển duy nhất, giúp các nhà quản trị dễ dàng cấu hình và giám sát toàn bộ hệ thống mà không cần can thiệp vào từng thiết bị riêng lẻ. Điều này đặc biệt hữu ích trong những môi trường phức tạp như trung tâm dữ liệu hoặc mạng doanh nghiệp lớn.
    - Theo như báo cáo thống kê của Cisco, 64% các tổ chức đã áp dụng SDN trong trung tâm dữ liệu, 58% trong mạng WAN và 40% trong mạng truy cập. (Cisco, n.d.)
    - Ví dụ thực tế: Airbus Helicopters là một bộ phận sản xuất tại Airbus, với trụ sở chính tại Pháp, là một trong những công ty lớn nhất trên thị trường, xét về cả sản lượng và doanh thu. Họ đã triển khai Cisco ACI, một giải pháp SDN, để kết nối nhiều toà nhà và gần 2000 người dùng với nhau. Airbus Helicopters đã thay thế khoảng 20% cơ sở hạ tầng hiện có, tương đương khoảng 43 switch truyền thống. Điều này giúp đơn giản hoá quản lý mạng và tăng tính linh hoạt, tăng khả năng giải quyết khối lượng công việc lớn của họ. (Anina Ot, 2022)

  Theo như những thống kê và ví dụ, việc quản lý mạng linh hoạt đã làm tăng khả năng thích ứng với các yêu cầu kinh doanh thay đổi nhanh chóng. Ngoài ra, cũng cải thiện được khả năng quan sát và kiểm soát hành vi mạng.
- **Tối ưu hoá hiệu suất và tài nguyên mạng**: SDN tối ưu hoá hiệu suất bằng cách phân bổ và tái phân bổ tài nguyên một cách linh động, giúp giảm thiểu tình trạng các tuyến đường bị tắc nghẽn và đảm bảo hiệu suất ổn định. Điều này đặc biệt quan trọng trong các mạng có lưu lượng cao hoặc yêu cầu theo thời gian thực.
    - Theo thống kê của Cisco có 64% tổ chức áp dụng SDN trong trung tâm dữ liệu, điều này cho thấy SDN đảm bảo được việc tối ưu hoá hiệu suất nên ngày càng phổ biến và được áp dụng nhiều.

- Ví dụ thực tế: Nghiên cứu từ MDPI đã đánh giá hiệu suất SDN trong mạng Wi-Fi quy mô lớn, nhiều trường hợp được đặt ra, như 1 điểm truy cập, 2 điểm truy cập, 3 điểm truy cập và 4 điểm truy cập của hai giao thức TCP và UDP. Dưới đây là hình minh hoạ băng thông TCP và UDP tại 1 điểm truy cập và 4 điểm truy cập.

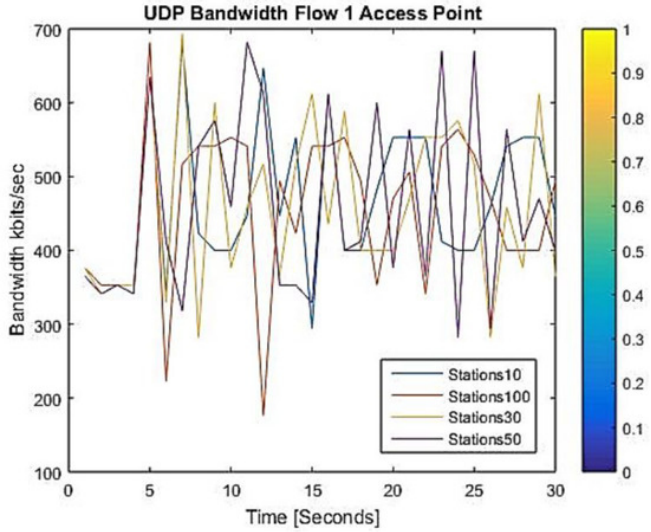


Hình 2.2: Băng thông TCP cho 1 điểm truy cập

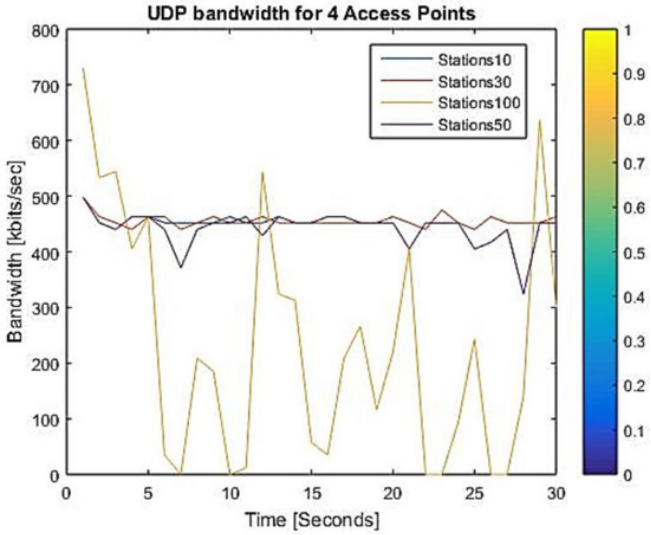


Hình 2.3: Băng thông TCP cho 4 điểm truy cập

Với 1 điểm truy cập, băng thông TCP đạt 440-498 kb/s cho từ 10 đến 50 thiết bị, nhưng giảm xuống còn khoảng 255-498 kb/s khi có 100 thiết bị. Tại trường hợp 4 điểm truy cập, giá trị băng thông cho 10, 30, 50 và 100 thiết bị lần lượt là 440-498 kb/s, 440-498 kb/s, 324-498 kb/s và 0-730 kb/s. (Ali et al., 2023)



Hình 2.4: Băng thông UDP cho 1 điểm truy cập



Hình 2.5: Băng thông UDP cho 4 điểm truy cập

Đối với UDP, mức độ dao động của băng thông là 282-694 kb/s đối với 10-50 thiết bị và 172-682 kb/s khi đạt 100 thiết bị tại 1 điểm truy cập. Trường hợp 4 điểm truy cập, giá trị băng thông tại 10-50 rất ổn định không có dao động mạnh, và khi có 100 thiết bị giá trị băng thông đạt 0-730 kb/s.

Nghiên cứu này minh hoạ cách SDN điều chỉnh hiệu suất dựa trên số lượng thiết bị và cấu hình mạng. Nó cũng có thể phân phối lưu lượng đến các điểm truy cập sao cho đạt được cân bằng tải, điều này rất quan trọng với trường hợp có mức thiết bị cao.

- **Tự động hoá và khả năng mở rộng cao**: Mô hình SDN hỗ trợ tự động hoá thông qua các API, cho phép lập trình và cấu hình mạng một cách tự động, làm giảm thiểu lỗi do con người gây ra. Ngoài ra, nó còn giúp mở rộng mạng dễ dàng mà không gây gián đoạn.
    - Đây là một vài số liệu thống kê nổi bật cho thấy sự trỗi dậy của SDN: Với 57% tổ chức dự kiến áp dụng SDN vào năm 2023. Khoảng 49% trung tâm dữ liệu toàn cầu dự kiến triển khai SDN vào năm 2024. Còn thị trường SDN toàn cầu dự kiến đạt 35,9 tỷ USD vào năm 2024, với tốc độ tăng trưởng CAGR là 37,9%. (Andersen & MoldStud Research Team, 2024)
    - Ví dụ thực tế: CloudSeeds là nhà cung cấp dịch vụ tư vấn và kỹ thuật đám mây từ năm 2013 tại Hamburg, Đức. Công ty giúp khách hàng mở rộng mạng lưới của mình để đạt được mục tiêu thông qua ảo hoá cơ sở hạ tầng và dịch vụ CNTT. Họ đã sử dụng Contrail để triển khai dịch vụ SDN, trong vài phút họ đã đưa máy chủ mới vào hoạt động hoặc có thể triển khai toàn bộ trung tâm dữ liệu mới cho khách hàng chỉ trong vài giờ thay vì vài tuần, đồng thời giảm chi phí nhờ vào khả năng tự động hoá và mở rộng của SDN. (Anina Ot, 2022)

SDN mang lại lợi ích vượt trội trong việc tự động hóa triển khai và cấu hình, điều này làm giúp giảm đáng kể sự phụ thuộc vào nhân sự kỹ thuật truyền thống, từ đó tiết kiệm chi phí vận hành và giảm thiểu sai sót cấu hình thủ công. Ngoài ra với kiến trúc linh hoạt, SDN cho phép mở rộng mạng nhanh chóng để đáp ứng nhu cầu tăng trưởng về người dùng, thiết bị và dịch vụ mà không làm gián đoạn hệ thống hiện tại.

- **Cải thiện bảo mật mạng**: Yếu tố an ninh mạng đã trở thành thành phần thiết yếu trong cấu trúc mạng của bất kì doanh nghiệp. SDN có thể đáp ứng được yêu cầu về bảo mật mà mạng truyền thống không thể làm được.
    - Trước hết, SDN cho phép dễ dàng tạo ra các mạng cô lập mà không bị giới hạn bởi phần cứng vật lý, giúp cách ly các ứng dụng và dịch vụ có mức nhạy cảm khác nhau. (Security Guidance Working Group, 2021)
    - Tường lửa của SDN có thể được áp dụng linh hoạt theo tiêu chí như thẻ (tag), thay vì chỉ dựa vào địa chỉ IP như các tường lửa vật lý truyền thống. (Security Guidance Working Group, 2021)
    - Một điểm mạnh khác là chính sách mặc định thường là từ chối, yêu cầu quản trị viên chủ động cấp quyền truy cập. Điều này giúp kiểm soát tốt hơn và hạn chế các lỗ hổng mặc định. (Security Guidance Working Group, 2021)
    - Ngoài ra, nhiều hình thức tấn công mạng như giả mạo ARP hay khai thác tầng thấp được loại bỏ mặc định nhờ kiến trúc ảo hoá, trong khi các gói tin có thể được mã hoá ngay khi đóng gói. (Security Guidance Working Group, 2021)
    - Cuối cùng, SDN cho phép tích hợp các chức năng bảo mật bổ sung một cách lịnh hoạt và tự nhiên trong hạ tầng mạng. (Security Guidance Working Group, 2021)

- Ví dụ thực tế: Từ nghiên cứu của Krzysztof Cabaj và Wojciech Mazurczyk đã trình bày cách sử dụng SDN để giảm thiểu mối đe doạ từ ransomware, cụ thể là CryptoWall. Họ đã đề xuất hai phương án giảm thiểu thời gian thực và thiết kế một hệ thống dựa trên SDN với giao thức OpenFlow, cho phép phản ứng kịp thời với các mối đe doạ mà không ảnh hưởng đáng kể đến hiệu suất mạng. (Cabaj & Mazurczyk, 2016)

- **Giảm chi phí:** SDN giúp giảm các nguồn chi phí thông qua tự động hoá, giảm nhu cầu nhân sự và sử dụng phần cứng giá rẻ. Điều này giúp các tổ chức tiết kiệm cả chi phí vốn và chi phí vận hành. Dưới đây là bảng so sánh các chi phí khi triển khai mô hình mạng giữa mô hình truyền thống và SDN.

Hạng mục chi phí	**Mạng truyền thống**	**Mạng SDN**
Chi phí đầu tư ban đầu (CAPEX)	Chi phí cao do sử dụng thiết bị phần cứng chuyên dụng của những hãng lớn và phần mềm độc quyền từ Cisco, Juniper,…	Thấp hơn nhờ sử dụng thiết bị phần cứng phổ thông và phần mềm mã nguồn mở. Tiết kiệm từ 10-40%. (Mohsin, 2022)
Chi phí vận hành (OPEX)	Cao do yêu cầu cấu hình thủ công, nhân lực IT đông và tốn thời gian vận hành.	Quản lý tập trung qua SDN controller giúp giảm nhân lực và sai sót. Tự động hoá giúp tiết kiệm 10-80% chi phí vận hành. (Mohsin, 2022)
Chi phí mở rộng hệ thống	Mỗi lần mở rộng cần bổ sung thiết bị chuyên	Dễ dàng mở rộng thông qua cấu hình phần mềm,

	dụng, cấu hình phức tạp và phụ thuộc vào nhà cung cấp.	không phụ thuộc nhiều vào phần cứng, giảm đáng kể chi phí mở rộng và thời gian triển khai.
Tổng chi phí sở hữu (TCO)	TCO cao do cộng dồn từ CAPEX và OPEX lớn.	Giảm tổng thể từ 20-80% nhờ tiết kiệm ở cả CAPEX, OPEX và mở rộng linh hoạt.(Mohsin, 2022)
Thời gian hoàn vốn (ROI)	ROI chậm hơn do đầu tư ban đầu lớn và chi phí duy trì cao.	ROI nhanh hơn nhờ chi phí giảm và hiệu suất quản lý tăng.

Bảng 2.2: Bảng so sánh mức chi phí triển khai

- Thống kê về chi phí vận hành: 5 ứng dụng SDN giúp tiết kiệm chi phí hiệu quả nhất gồm: Wifi Offload/Video Redirect, Cloud RAN, Local Breakout, Metro Aggregation và Small Cell, với tổng mức tiết kiệm được dự đoán lên đến 8,959 triệu USD vào năm 2017. Về khu vực, Châu Á – Thái Bình Dương đạt mức tiết kiệm lớn nhất với 5,619 triệu USD, nhờ triển khai mạnh mẽ mạng toàn IP. (Tellabs, 2014)
- Ví dụ thực tế: Thành phố Avondale, Arizona đã sử dụng Vmware NSX để tận dụng dung lượng dư thừa, tránh chi phí mua firewall và bảo trì cân bằng tải. Trong 3-5 năm tới, thành phố sẽ dự kiến thay thế switch và router bằng các thiết bị giá rẻ để tiếp tục tiết kiệm chi phí. (Zurier, 2015)

Với những so sánh và thống kê, ví dụ đã thể hiện rõ tiềm năng trong việc giảm chi phí và tối ưu vận hành mạng của SDN.

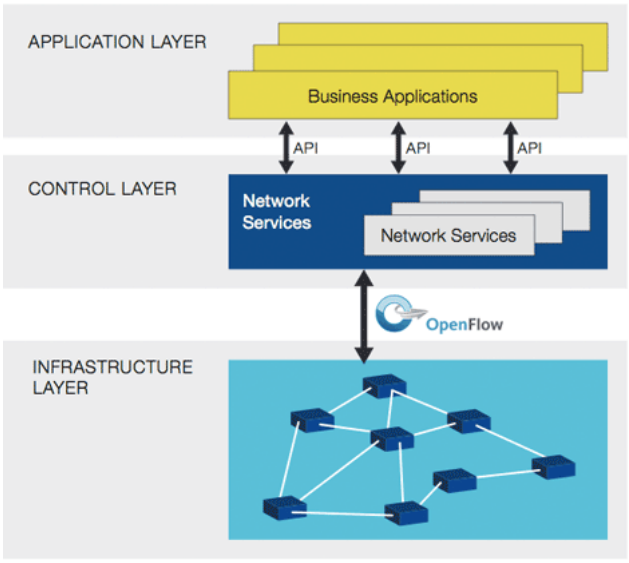
- **Hỗ trợ điện toán đám mây và ảo hoá**: Mạng SDN là nền tảng quan trọng cho điện toán đám mây và ảo hoá, cho phép tạo các mạng ảo linh hoạt và quản lý hiệu các tài nguyên đám mây.
    - Tuy không có số liệu thống kê cụ thể về tỷ lệ áp dụng SDN trong điện toán đám mây, nhưng SDN nhận được nhiều sự công nhận rằng đây là một giải pháp lý tưởng cho các môi trường đám mây.
    - Ví dụ thực tế: Một tổ chức tài chính toàn cầu đã triển khai SDN để giải quyết vấn đề hạ tầng mạng cứng nhắc, không đáp ứng được nhu cầu kinh doanh thay đổi nhanh chóng. SDN đã giúp họ đơn giản hoá cơ sở hạ tầng, cho phép triển khai ứng dụng nhanh chóng, vừa nâng cao bảo mật vừa giảm chi phí. Điều này cho phép họ có một mạng lưới linh hoạt và an toàn, hỗ trợ các mục tiêu kinh doanh trong môi trường đám mây. (Alibaba Cloud, 2024)

Với những lợi ích mà SDN đem lại, điều này chứng minh rằng đây chính là nền tảng vững chắc cho các hệ thống mạng hiện đại. Đặc biệt trong bối cảnh ngày càng phát triển mạnh mẽ của điện toán đám mây, IoT và các yêu cầu bảo mật ngày càng khắc khe.

2.2 Kiến trúc SDN

2.2.1 Giới thiệu kiến trúc

Kiến trúc SDN là mô hình mạng tiên tiến, hiện đại, trong đó lớp điều khiển được tách biệt khỏi lớp vận chuyển dữ liệu và sử dụng bộ điều khiển trung tâm để quản lý toàn bộ mạng. Kiến trúc SDN bao gồm ba lớp chính: Lớp ứng dụng (Application Layer), lớp điều khiển (Control Layer) và lớp hạ tầng (Infrastructure Layer). Các lớp sẽ liên kết với nhay thông qua giao thức OpenFlow hoặc các API. Sự phân chia này loại bỏ sự phụ thuộc vào phần cứng cố định, cung cấp tính linh hoạt và khả năng thích ứng với các yêu cầu kinh doanh phức tạp, đặc biệt trong các trung tâm dữ liệu, mạng doanh nghiệp và điện toán đám mây.



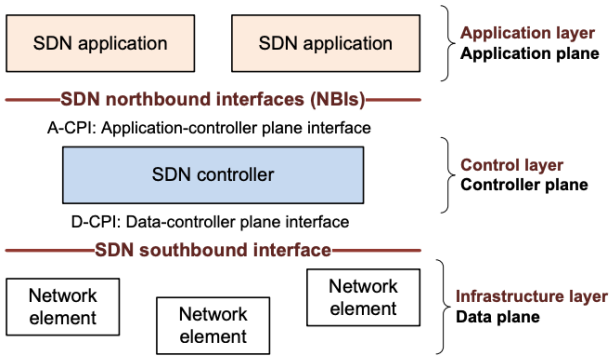
Hình 2.6: Kiến trúc SDN

- **Lớp ứng dụng**: Đây là nơi tập trung các ứng dụng mạng phục vụ cho việc kiểm soát, tối ưu và bảo mật hệ thống. Các ứng dụng phổ biến thường được các tổ chức, doanh nghiệp sử dụng như hệ thống phát hiện xâm nhập (IDS/IPS), tường lửa, cân bằng tải, công cụ giám sát, phân tích lưu lượng mạng,… Điểm khác biệt so với kiến trục mạng truyền thống là thay vì phải sử các thiết bị phần cứng chuyên dụng cho từng chức năng, thì với SDN các chức năng tại lớp này có thể triển khai dưới dạng phần mềm, chạy trên các máy chủ thông thường.
- **Lớp điều khiển**: Lớp này được xem là một bộ não trung tâm của kiến trúc SDN, nơi mà bộ điều khiển (Controller) vận hành để quản lý toàn bộ mạng. Như hình minh hoạ, lớp này được thiết kế nằm ở vị trí trung gian giữa lớp ứng dụng và lớp hạ tầng, nên có hai chức năng chính:
    - Cung cấp giao diện lập trình ứng dụng (API) cho lớp ứng dụng để các chính sách mạng có thể định nghĩa và triển khai.
    - Thực hiện giao tiếp với các thiết bị mạng ở lớp hạ tầng bằng giao thức OpenFlow để thu thập thông tin và áp dụng các chính sách điều khiển.

Thay vì mỗi thiết bị mạng tự quyết định tuyến đi và xử lý dữ liệu như trong mô hình truyền thống, thì việc ra quyết định được thực hiện tập trung tại lớp này. Nhờ đó mà hệ thống trở nên đồng bộ hơn, dễ quản lý hơn và có khả năng thích ứng nhanh với các thay đổi hoặc sự cố trong mạng.

- **Lớp hạ tầng**: Gồm các thiết bị như switch, router và các thiết bị mạng khác, đóng vai trò tiếp nhận và chuyển tiếp dữ liệu. Các thiết bị này không còn đảm nhiệm việc tự quyết định tuyến đi hay xử lý mạng như các thiết bị trong mô hình mạng truyền thống, mà hoạt động theo sự chỉ thị từ lớp điều khiển thông qua giao thức OpenFlow. Đồng thời, các thiết bị này cũng có khả năng gửi thông tin cho bộ điều khiển về tình trạng lưu lượng và hoạt động, tạo điều kiện cho việc giám sát và tối ưu hệ thống theo thời gian thực. Với mô hình này, các doanh nghiệp có thể sử dụng các thiết bị phổ thông thay vì đầu tư vào thiết bị cao cấp nhiều chức năng, từ đó giảm chi phí tổng thể nhưng vẫn đảm bảo hiệu năng toàn mạng.

Ngoài ra trong kiến trúc SDN, hai thành phần đóng vai trò quan trọng trong việc liên kết giữa các lớp là Northbound Interface (NBI) và Southbound Interface (SBI). Hai thành phần này giúp tạo ra sự tách biệt rõ ràng giữa ba lớp với nhau, nhưng vẫn đảm bảo tính linh hoạt của hệ thống.



Hình 2.7: Kiến trúc SDN (2)

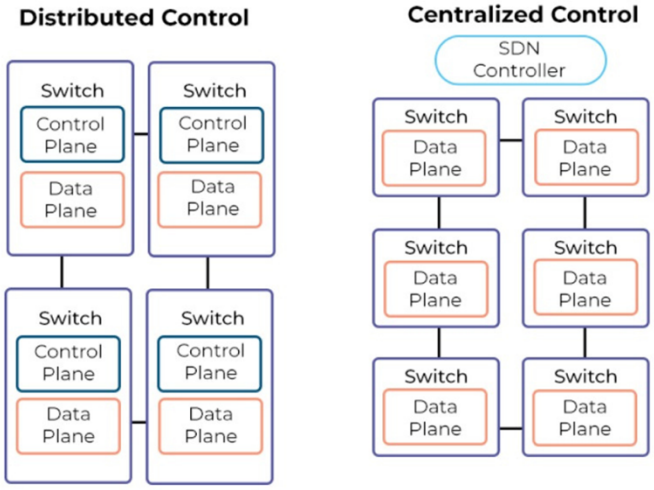
(Open Networking Foundation, 2014)

- **Northbound Interface**: Là giao diện kết nối giữa lớp điều khiển và lớp ứng dụng. Nó cho phép các ứng dụng truy xuất dữ liệu mạng, thống kê lưu lượng và tương tác với bộ điều khiển. Nhờ có Northbound Interface, các nhà quản trị và nhà phát triển có thể dễ dàng xây dựng các ứng dụng mạng tuỳ chỉnh mà không cần can thiệp vào cơ sở hạ tầng.
- **Southbound Interface**: Giao diện southbound phổ biến nhất là OpenFlow cho phép controller điều khiển hành vi của switch bằng cách cài đặt các luồng trong bảng định tuyến. Ngoài ra, các giao thức như ForCES (mô hình hoá bằng các khối chức năng logic - LFB), SoftRouter (cho phép phân bổ điều khiển động), PCE và LISP cũng là những lựa chọn tiềm năng. So với OpenFlow, ForCES có tính linh hoạt cao hơn trong khi SoftRouter nhấn mạnh độ tin cậy trong điều khiển bạn như BGP. (Braun & Menth, 2014)

Sự tách biệt rõ ràng và linh hoạt giữa Northbound Interface và Southbound Interface chính là điểm mạng của SDN so với mạng truyền thống. Nó giúp giảm sự phụ thuộc vào các chức năng cá nhân của từng thiết bị vật lý và cho phép phát triển các giải pháp mạng thông minh, tự động hoá cao.

2.2.2 Cách hoạt động của SDN

Vì mô hình SDN mang đến một cách tiếp cận mới trong thiết kế và vận hành mạng bằng cách phân tách chức năng điều khiển ra khỏi phần chức năng vận chuyển dữ liệu. Thay vì mỗi thiết bị mạng tự đưa ra quyết định điều khiển, SDN tập trung toàn bộ quyền điều phối vào một hệ thống điều khiển trung tâm (Theo minh hoạ hình 2.8).

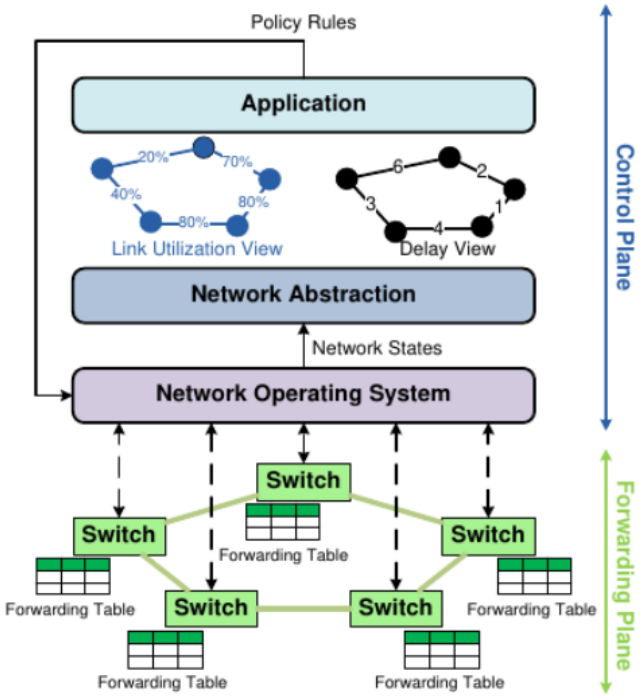


Hình 2.8: Cách hoạt động của SDN

Trong phần này, sẽ tìm hiểu cách mà hai thành phần chính của SDN phối hợp để đảm bảo hoạt động của toàn bộ hệ thống mạng, quá trình này được minh hoạ theo (Hình 2.9).

- **Forwarding Plane (Data Plane)**: Phần vận chuyển dữ liệu này bao gồm các switch đơn giản. Chức năng chính của các thiết bị chuyển mạch này là chuyển tiếp các gói tin dựa trên chính sách định tuyến do phần điều khiển chỉ định. Để thực hiện vai trò này, mỗi switch duy trì một bảng chuyển tiếp (forwarding table), trong bảng này chứa các quy tắc do phần điều khiển cài đặt. Các bảng có ba trường chính:
    - Trường mẫu: Xác định các gói tin thuộc một luồng cụ thể dựa trên tiêu đề.
    - Trường đếm: Ghi lại số lần quy tắc được áp dụng.
    - Trường hành động: Quy định cách xử lý gói tin, như là chuyển tiếp, loại bỏ hoặc gửi về bộ điều khiển.
- **Control Plane**: Phần điều khiển được xem như là bộ não của mạng, chịu trách nhiệm giám sát toàn bộ hệ thống, đưa ra các quyết định định tuyến và lập trình cho mạng vật lý cách thức hoạt động. Nó gồm ba lớp chính:

- Lớp hệ điều hành mạng (Network Operating System): Kết nối với các switch, thu thập thông tin trạng thái như độ trễ, kết nối và mức sử dụng liên kết.
- Lớp trừu tượng hoá mạng (Network Abstraction): Xử lý dữ liệu thu thập được để tạo ra các khung để có cái nhìn tổng thể về mô hình mạng, ví dụ như đồ thị mạng.
- Lớp ứng dụng (Application): Dựa vào các khung đã được tạo để chạy thuật toán tìm chính sách định tuyến tối ưu, sau đó gửi các quy tắc chuyển tiếp xuống lớp điều hành mạng để cấu hình các switch.



Hình 2.9: Mô tả hoạt động của SDN

(Dabbagh et al., 2015)

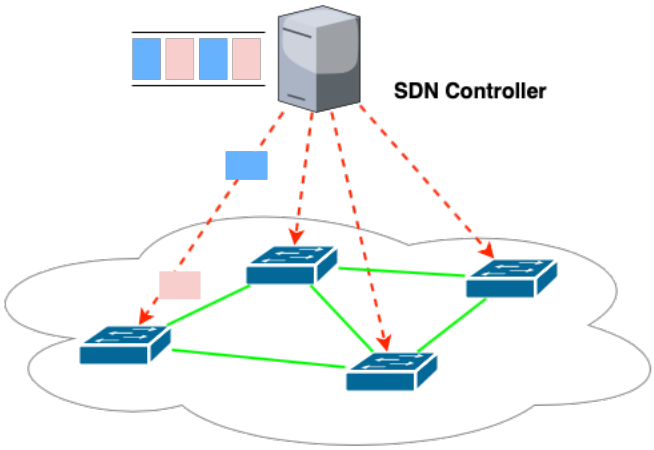
CHƯƠNG 3 – THÁCH THỨC VÀ GIẢI PHÁP TRONG HIỆN THỰC SDN

3.1 Các thách thức trong hiện thực

Mô hình SDN là một kiến trúc hiện đại, mang lại nhiều lợi ích thiết thực cho doanh nghiệp như khả năng quản lý linh hoạt, tự động hóa và tối ưu hóa hiệu suất mạng, nhưng trên thực tế, SDN vẫn chưa được triển khai rộng rãi ở quy mô toàn cầu. Mặc dù tiềm năng lớn, mô hình này đang phải đối mặt với nhiều thách thức khi áp dụng vào môi trường thực tế, bao gồm hiệu suất và độ trễ, khả năng mở rộng, bảo mật, độ tin cật, khả năng tương thích với hạ tầng cũ, và giao diện cấp thấp. Những khó khăn sẽ được phân tích cụ thể trong môi trường doanh nghiệp, trung tâm dữ liệu, ISP và môi trường điện toán đám mây trong phần này.

3.1.1 Hiệu suất và độ trễ

Theo như đã tìm hiểu về kiến trúc của SDN, thì việc tách biệt phần điều khiển ra khỏi phần dữ liệu giúp SDN linh hoạt trong quản lý nhưng nó gây ra độ trễ cho mô hình mạng. Cụ thể, tất cả các quyết định điều khiển mạng đều do bộ điều khiển (controller) đảm nhận (hình minh hoạ 1), mỗi luồng dữ liệu mới phải được gửi đến controller để cài đặt đường đi, hay còn được gọi là flow setup, và độ trễ phản hồi của controller sẽ cộng vào thời gian vận chuyển gói tin. Như một bài báo cho thấy, với một mô hình mạng có 100 switch có thể xảy ra các đợt tăng đột biến lên đến 10 triệu luồng/giây trong tình huống xấu nhất. Ngoài ra, thời gian trì hoãn để chờ controller thiết lập luồng là khoảng 10ms và nó có thể bị tăng thêm 10% độ trễ cho phần lớn luồng mới trong mạng, đặc biệt là với các luồng ngắn hạn (short-lived) luôn chiếm phần lớn lưu lượng. (Tootoonchian et al., 2012). Bởi vì controller đóng vai trò là bộ não của mô hình mạng, nên khi nhận được số lượng yêu cầu khổng lồ nó dễ trở thành điểm tắc nghẽn gây nghẽn hệ thống. Nhìn chung thì SDN mang lại khả năng tối ưu và kiểm soát tập trung nhưng đổi lại là độ trễ khởi tạo luồng cao và nguy cơ quá tải bộ điều khiển trong các mạng lớn.



Hình 3.1: Controller phân phối các gói tin

❖ Trong mạng doanh nghiệp

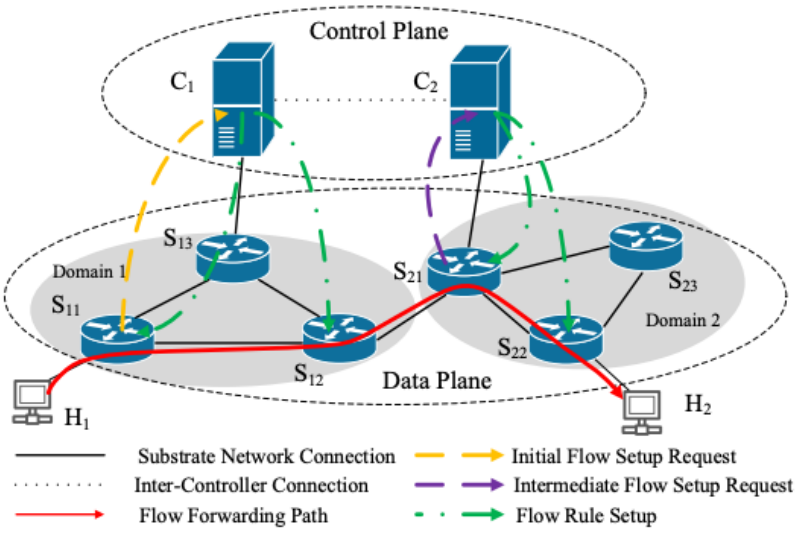
- Độ trễ khi thiết lập luồng mới: Mỗi khi thiết bị phát hiện luồng dữ liệu gặp lần đầu, nó phải gửi gói tin đến controller, chờ xử lý và thiết lập tuyến đi cho gói tin này. Mặc dù số lượng luồng trong môi trường doanh nghiệp không bằng trung tâm dữ liệu, nhưng độ trễ xử lý vẫn gây gián đoạn tạm thời cho các ứng dụng.
- Số lượng bộ điều khiển không lớn nhưng vẫn cần để ý: Ở quy mô doanh nghiệp, số lượng switch và yêu cầu còn khiêm tốn nên một controller có thể xử lý tốt. Tuy nhiên, nếu mở rộng mô hình thêm nhiều chi nhánh hoặc nhiều người dùng liên tục kết nối, controller cũng có thể bị quá tải và làm tăng độ trễ xử lý.

❖ Trong trung tâm dữ liệu

- Lưu lượng luồng khởi tạo dày đặc: Nghiên cứu cho thấy trong một cụm gồm 1500 máy chủ trung bình có khoảng 100,000 luồng mới mỗi giây, và có thể lên đến 10 triệu luồng/giây trong trường hợp tệ nhất. Trong khi đó, các controller ban đầu như NOX chỉ xử lý khoảng 30,000 luồng/giây với độ trễ dưới 10ms. Mặc dù đã cải tiến, NOX-MT đa luồng đạt 1,6 triệu yêu cầu trên mỗi giây với khoảng 2ms độ trễ phản hồi

(Tootoonchian et al., 2012), nhưng với mức độ nhập luồng khổng lồ của Data Center vẫn có thể vượt ngoài tầm xử lý nếu không dùng nhiều controller. Kết quả là một số gói ban đầu có thời gian chờ xử lý lâu, làm giảm hiệu năng tổng thể và gây ra ảnh hưởng cho các ứng dụng thời gian thực.

- Tương tác đa bộ điều khiển: Các Data Center lớn thường triển khai nhiều controller cho từng phân vùng hoặc từng tầng. Một luồng dữ liệu có thể đi qua nhiều bộ điều khiển, điều này nghĩa là phải gửi yêu cầu lập luồng đến các controller khác nhau. Quá trình xử lý gói tin tại mỗi controller sẽ cộng dồn độ trễ: switch S_{11} tại Domain 1 gửi gói tin đến C_1, khi có thông tin chỉ định đường đi thì chuyển đến Domain 2. Switch S_{21} gửi yêu cầu đến C_2 để đợi xử lý luồng di chuyển, (minh hoạ theo hình 3.2). Kết quả là độ trễ tăng lên, đặc biệt là với nhiều controller phân tán rộng. (He et al., 2017)



Hình 3.2: Luồng xử lý gói tin trong mô hình đa bộ điều khiển

(He et al., 2017)

❖ Trong môi trường ISP

- Đỗ trễ đường truyền điều khiển: Do phạm vi địa lý lớn, khoảng cách giữa router và controller có thể lớn làm tăng đáng kể RTT của các gói Paket-In và Flow-Mod. RTT là thời gian một gói tin mất để đi từ điểm gửi đến điểm nhận và quay lai, đây là một chỉ số quan trọng để đo lường độ trễ trong mạng. Thậm chí nếu dùng bộ điều khiển cục bộ, controller phải đồng bộ với controller khác để duy trì cái nhìn toàn cục, điều này có thể dẫn đến overhead. Như nghiên cứu chỉ ra rằng "độ trễ xử lý của controller và chi phí phản hồi trực tiếp ảnh hưởng đến độ trễ mạng" (Zhou & Tan, 2024), điều này có nghĩa là bất kỳ chậm trễ nào ở phần điều khiển cũng sẽ đều làm chậm việc truyền dữ liệu.
- Độ phức tạp điều phối đa miền: Mạng ISP thường được chia thành nhiều domain hành chính. Một luồng muốn đi từ miền này sang miền khác thường phải đi qua nhiều switch hoặc master khác nhau, như ở phần trung tâm dữ liệu. Ngoài ra, nếu có nhiều controller, mô hình mạng cần các cơ chế đồng bộ phức tạp, như northbound, inter-controller. Mỗi yêu cầu xử lý bổ sung giữa các controller có thể làm tăng độ trễ end-to-end.
- Khối lượng lưu lượng cao và khả năng đóng gói lưu lượng: Với môi trường của ISP phải xử lý lượng lớn gói tin, bao gồm cả gói lớn và gói nhỏ. Khi mạng truyền thống có thể gộp lưu lượng qua BGP hoặc MPLS, thì SDN phải quản lý từng luồng chi tiết, do đó mà tăng đáng kể lượng công việc của CPU trên thiết bị. Hậu quả gây ra là controller phải xử lý hàng triệu luồng, dễ gây tắc nghẽn.

❖ Trong điện toán đám mây:

Trong điện toán đám mây, SDN được sử dụng rộng rãi để ảo hoá mạng, như OpenStack, Neutron, VMware NSX, Azure VNets,…Môi trường này có đặc điểm là lượng client và máy ảo cực lớn với nhiều luồng nhỏ và tính đa

nhiệm cao, do đó không thể tránh khỏi những thách thức về hiệu suất và độ trễ. Nổi bật nhất là vấn đề về quy mô và đa chia luồng: Các dịch vụ đám mây công cộng vận hành hàng trăm ngàn, trăm triệu máy ảo động. Một thử nghiệm với 5000 mạng trong OpenStack/Neutron và kích hoạt VM. Họ đã quản lý để có được tốc độ khoảng 10 VM/giây (tức 600 VM/phút) cao hơn gấp đôi so với thí nghiệm cũ của họ (Stiliadis, 2014). Điều này có nghĩa SDN phải xử lý hàng trăm quy trình mạng đồng thời, điều này dễ gây tắc nghẽn các hàng đợi khi đang chờ thiết lập chính sách.

3.1.2 Khả năng mở rộng

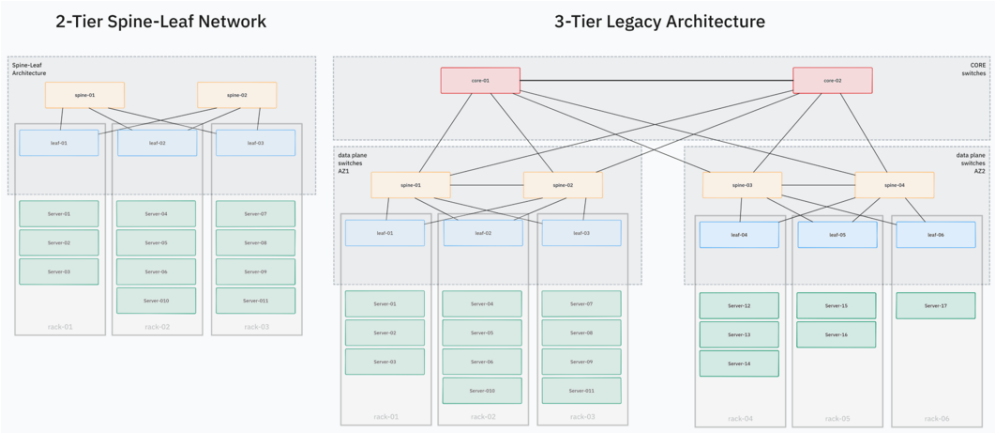
Khả năng mở rộng là khả năng của mạng để xử lý lượng công việc tăng lên hoặc mở rộng để đáp ứng nhu cầu. Đương nhiên khả năng này của SDN cũng bị hạn chế bởi kiến trúc tập trung, một controller duy nhất không thể xử lý số lượng lớn switch trong mạng quy mô lớn. Việc đồng bộ hoá giữa nhiều controller trong mô hình phân tán cũng góp phần gây ra độ trễ trong mô hình.

- ❖ Môi trường doanh nghiệp:
    - Giới hạn bộ nhớ luồng trên switch: Trong SDN, OpenFlow sử dụng TCAM để lưu trữ và khớp các quy tắc định tuyến trong bảng luồn của switch, thường chỉ lưu trữ tối đa khoảng 1000-2000 luồng. Khi số lượng luồng vượt quá ngưỡng này, controller phải liên tục cài đặt và xoá luồng, điều này gây quá tải cho Control Plane và làm tăng độ trễ xử lý. (Modali, 2016). Ngoài ra, việc chỉ có một controller cũng làm giảm tính dự phòng trong doanh nghiệp, khi gặp sự cố, toàn bộ hệ thống mạng có thể ảnh hưởng nghiêm trọng.
    - Tích hợp với hệ thống mạng cũ: Hầu hết doanh nghiệp vẫn sử dụng thiết bị mạng và phần mềm cũ chưa hỗ trợ SDN, dẫn đến vấn đề tương thích khi chuyển đổi. Việc tích hợp SDN vào hạ tầng legacy thường phức tạp và tốn kém vì các thiết bị cũ không được thiết kế để nhận lệnh điều khiển

từ xa. Do đó doanh nghiệp thường phải áp dụng SDN một cách từng phần, khiến quá trình mở rộng quy mô mạng chậm hơn và ít hiệu quả hơn.

❖ Trung tâm dữ liệu:

- Thiết kế overlay, underlay phức tạp: Mạng SDN ở trung tâm dữ liệu thường được xây dựng như một lớp mạng ảo (overlay) trên hạ tầng vật lý (underlay). Nếu overlay không tương thích với kiến trúc vật lý, hiệu suất mạng sẽ giảm sút. Ví dụ, kiến trúc spine-leaf đã chứng minh giúp mở rộng dễ dàng nhờ kết nối đơn giản giữa các tầng.



Hình 3.3: Mô hình Spine-Leaf

(Kafazov, 2024)

SDN ảo phải được thiết kế hợp lý dựa trên topology vật lý để đạt hiệu năng và khả năng mở rộng cao nhất. Nếu không, các đường dẫn ảo có thể gây tắc nghẽn hoặc không tận dụng hết băng thông của dưới tầng. (Kafazov, 2024)

❖ Trong môi trường ISP

- Quy mô rất lớn và hạ tầng di sản: Mạng ISP có quy mô toàn cầu với hàng ngàn thiết bị, thường phát triển qua nhiều năm và tích hợp nhiều loại thiết bị và giao thức khác nhau. Nếu như thay thế toàn bộ bằng mô hình

SDN mới thì việc đó là không khả thi về chi phí và rủi ro, thay vào đó ISP phải kết hợp SDN từ từ. Nghiên cứu cho thấy kiến trúc SDN "thuần" không thể áp dụng trực tiếp cho mạng ISP đã xây dựng lâu dài và đa dạng thiết bị. Do đó các giải pháp SDN trong mạng ISP thường phải tương thích ngược và chỉ triển khai một phần, dẫn đến việc hạn chế khả năng mở rộng trên quy mô toàn mạng.

- Yêu cầu SLA nghiêm ngặt: Khả năng mở rộng của SDN trong môi trường ISP gặp thách thức lớn do yêu cầu về thoả thuận mức dịch vụ (SLA) nghiêm ngặt về tính sẵn sàng, độ trễ thấp và băng thông cố định cho doanh nghiệp. Mọi luồng dữ liệu (như VoIP, video trực tuyến) đều cần xử lý thời gian thực và băng thông phải được duy trì ổn định. Điều này làm tăng gánh nặng cho controller, vì nó phải xử lý nhanh và đảm bảo các yêu cầu nghiêm ngặt.
- Điều khiển phân tán và điểm nghẽn: Do quy mô khổng lồ, các ISP thường triển khai nhiều bộ điều khiển phân tán để tránh tắc nghẽn. Tuy nhiên, việc đồng bộ trạng thái giữa các controller cũng là thách thức lớn. Nếu chỉ dùng một controller tập trung, nó sẽ khó đáp ứng được hàng triệu lượt sự kiện luồng mỗi giây, dễ gây quá tải và mất khả năng mở rộng. Hệ quả là hệ thống SDN phải thiết kế theo mô hình phân tán phức tạp, tăng độ trễ đồng bộ và giảm hiệu năng tổng thể.

❖ Trong điện toán đám mây

- Đa thuê và ảo hoá quy mô lớn: Môi trường đám mây công cộng phục vụ hàng trăm đến hàng nghìn khách hàng cùng chia sẻ tài nguyên. Mỗi người thường có mạng ảo riêng, nên số lượng VM và mạng overlay rất lớn. Khi số VM tăng lên, kiến trúc SDN (ví dụ OpenStack Neutron) bị đặt dưới áp lực lớn. Controller phải quản lý đồng thời nhiều mạng ảo, bảng định tuyến và chính sách cách ly giữa các khách hàng. Điều này

tăng mạnh khối lượng công việc cho control plane và dễ gây nghẽn khi muốn mở rộng quy mô. (Kafazov, 2024)

- Phương thức overlay dễ thất bại: SDN trong đám mây thường dùng VLAN/VXLAN để tạo mạng ảo trên nền hạ tầng vật lý. Tuy nhiên, phương pháp này rất dễ bị lỗi rộng. Một VM bất thường có thể làm sập toàn bộ mạng ảo, tạo "đổ vỡ lớn" (blast radius) rất cao. Ngoài ra, mỗi mạng ảo mới đòi hỏi một VLAN tương ứng trên hạ tầng vật lý, khi quá nhiều VLAN được tạo, sẽ vượt ngưỡng tối đa của switch và gây tắc nghẽn. Những yếu tố này làm giảm khả năng mở rộng thực tế của SDN trong môi trường đám mây, vì giải pháp mạng ảo phụ thuộc nặng nề vào tầng vật lý và tính ổn định kém.

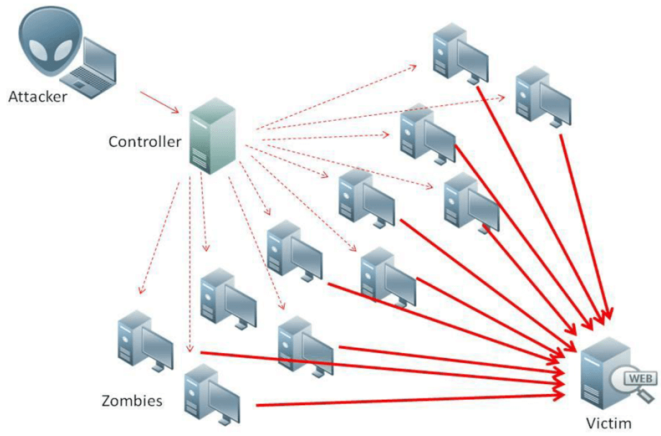
3.1.3 Thách thức về bảo mật

Vì controller đóng vai trò trung tâm, chịu trách nhiệm quản lý và điều khiển toàn bộ hoạt động mạng. Tuy nhiên, kiến trúc tập trung này cũng mang lại rủi ro về bảo mật, nó vừa là lợi thế cũng vừa là yếu điểm lớn nhất của SDN. Nếu controller bị tấn công hoặc quá tải, toàn bộ mạng có thể bị gián đoạn nghiệm trọng, ảnh hưởng đến dịch vụ và hoạt động doanh nghiệp.

Cả trong môi trường doanh nghiệp, trung tâm dữ liệu, ISP hay điện toán đám mây thì cũng không tránh khỏi việc bị tấn công, điển hình là tấn công DDoS vào controller. DDoS – Distributed Denial of Service, là hoạt động làm chấm dứt hoặc gián đoạn các dịch vụ tại bộ điều khiển. Tấn công DDoS huy động số lượng lớn các máy ma gửi một lượng lớn gói tin đến controller. Ngoài ra, kẻ tấn công có thể khai thác việc thiếu giới hạn tốc độ xử lý gói tin ban đầu để làm tràn bộ nhớ hoặc gây nghẽn tại bộ điều khiển. Đặc biệt:

- Trong doanh nghiệp, DDoS có thể khiến các dịch vụ nội bộ bị gián đoạn.
- Trong trung tâm dữ liệu và ISP, tấn công vào controller có thể làm ảnh hưởng đến toàn bộ khách hàng đang sử dụng dịch vụ và hạ tầng ảo hoá.

- Trong đám mây, DDoS có thể lan rộng giữa các vùng và làm gián đoạn kết nối đa đám mây.



Hình 3.4: Minh hoạ tấn công DDoS

Bên cạnh khó khăn chung là bị tấn công bằng DDoS, thì mỗi môi trường còn có những khó khăn riêng biệt trong mô hình SDN này. Dưới đây là phân tích các rủi ro chính khi triển khai SDN trong từng môi trường:

❖ Môi trường doanh nghiệp:

- Mục tiêu tấn công vào bộ điều khiển: Bộ điều khiển SDN là "đầu não" của toàn bộ mạng, nếu bị xâm phạm, kẻ tấn công có thể chi phối mọi luồng và chính sách mạng. Thực tế cho thấy bộ điều khiển thường là mục tiêu hàng đầu của hacker vì nó là "điểm yếu trung tâm" và "điểm thất bại" duy nhất của mạng SDN. (Chickowski, 2018)
- Tấn công giả mạo nội bộ (Spoofing): Môi trường LAN doanh nghiệp dễ xảy ra các tấn công giả mạo như ARP spoofing hoặc giả mạo địa chỉ thiết bị. Ví dụ, kẻ xấu có thể gửi các gói ARP giả để chiếm quyền điều khiển luồng nội bộm gây tê liệt mạng.
- Sai sót trong cấu hình: SDN cho phép lập trình mạng rất linh hoạt nhưng đồng thời dễ dẫn đến lỗi cấu hình. Chẳng hạn lỗi khai báo quy tắc hoặc

chính sách sai có thể vô tin mở rộng vùng tin cậy, cho phép truy cập chéo giữa các phòng ban hoặc thậm chí lộ dữ liệu khách hàng.

❖ Trung tâm dữ liệu

- Lỗ hổng giao thức OpenFlow: Giao thức OpenFlow theo thiết kế không bắt buộc chứng thực thiết bị nên có thể bị tấn công. Tin tặc có thể lợi dụng quá trình mà controller không xác thực switch để giả mạo thiết bị, gửi các thông báo điều khiển giả, dẫn đến DoS hoặc thao túng bản đồ mạng. (Myerson, 2018)
- Kênh điều khiển không an toàn: Kết nối southbound giữa controller và thiết bị phải dùng mã hoá. ONF khuyến cáo dùng TLS/DTLS với chứng chỉ để bảo vệ kênh này (ONF Solution Brief, 2013). Nếu kênh điều khiển không được mã hoá hoặc xác thực kém, kẻ xấu có nghe lén hoặc chèn điều khiển lên thiết bị, gây gián đoạn hiệu năng và hỏng tính toàn vẹn mạng. (ONF Solution Brief, 2013).
- Kiến trúc ảo hoá phức tạp: Trung tâm dữ liệu SDN thường có môi trường ảo hoá quy mô lớn với nhiều máy ảo, luồng di chuyển liên tục và các link tốc độ cao, sự phức tạp này đòi hỏi hệ thống bảo mật linh hoạt. Như tài liệu của ONF chỉ ra, các công nghệ bảo mật cố định thường không theo kịp môi trường Data Center, gây khó khăn trong quản lý và mở rộng. (ONF Solution Brief, 2013)

❖ Trong môi trường ISP:

- Kiến trúc phân tán quy mô lớn: ISP thường triển khai nhiều bộ điều khiển và vùng mạng rộng. Do đó, việc đồng bộ chính sách và dữ liệu trạng thái giữa các controller phân tán và thách thức lớn. Đặc biệt, các giải pháp bảo mật dựa trên mô hình một bộ điều khiển trung tâm không dễ áp dụng cho mạng ISP đa miền. Nghiên cứu cho thấy trong trường hợp controller

phân tán, việc ngăn chặn các cuộc tấn công DDoS quy mô lớn gặp khó khăn đáng kể.

- Tương tác với hạ tầng mạng truyền thống: Mạng ISP còn vận hành các giao thức định tuyến truyền thống như BGP, MPLS,…Việc tích hợp SDN với các giao thức này tạo nên rủi ro riêng, nếu tại bước điều phối luồng của SDN xung đột với các router, có thể xảy ra vòng lặp định tuyến. ISP phải xây dựng cơ chế xác nhận và đồng bộ an toàn giữa các lớp điều khiển mới và truyền thống để tránh lỗi giao tiếp.

❖ Trong điện toán đám mây

- Phân vùng và cách ly khách hàng: Môi trường đám mây thường nhiều khách hàng sử dụng dịch vụ, do đó phải đảm bảo cách ly tuyệt đối giữa các khách hàng, đây là điều bắt buộc. Bất kỳ sự cố giả mạo hoặc lỗi cấu hình nào cũng có thể lộ thông tin giữa các khách hàng. Vì thế, mà cơ chế cách ly khách hàng trong SDN, như qua overlay và định tuyến mạng ảo phải được thiết kế cẩn trọng.
- Giao diện và API quản lý: Đám mây sử dụng nhiều công cụ điều phối tự động như OpenStack, Kubernetes, API public cloud. Bất kỳ lỗ hổng nào trong API quản lý hoặc máy ảo quản lý cũng có thể dẫn đến kẻ thù kiểm soát luồng và cấu hình mạng. Ví dụ, nếu khoá API của controller bị lộ, kẻ tấn công có thể tạo luồng mới hoặc thay đổi chính sách của nhiều khách hàng cùng lúc. Các nhà cung cấp dịch vụ đám mây cần áp dụng thêm xác thực chặt chẽ cho mọi kênh quản lý SDN như NSA khuyến cáo (NSA, 2023).

3.1.4 Độ tin cậy

Một trong những yếu tố then chốt quyết định hiệu quả hoạt động của mạng SDN là độ tin cậy. Trong môi trường mạng động và quy mô lớn, việc đảm bảo mạng luôn duy trì hoạt động ổn định, không gián đoạn khi xảy ra sự cố là yêu cầu bắt buộc. Tuy nhiên,

do đặc tính tập trung hóa của SDN và sự phụ thuộc vào các controller, kiến trúc này cũng đối mặt với nhiều thách thức nghiêm trọng về độ tin cậy.

- ❖ Môi trường doanh nghiệp:
    - Điểm lỗi đơn của bộ điều khiển: Trong kiến trúc SDN doanh nghiệp, bộ điều khiển thường là thành phần tập trung quan trọng. Nếu chỉ sử dụng một bộ điều khiển duy nhất, thì khi nó gặp sự cố, toàn bộ mạng có thể mất khả năng điều khiển. Điều này khiến SDN trở thành điểm lỗi đơn (single point of failure), ảnh hưởng trực tiếp đến khả năng vận hành liên tục của mạng. (NSA, 2023, Dec 12)
    - Nguồn lực hạn chế: Nhiều doanh nghiệp nhỏ và vừa có nguồn lực CNTT hạn chế, dẫn đến việc khó khăn trong việc duy trì kiến trúc SDN dự phòng. Thiếu nhân lực am hiểu SDN và công cụ giám sát hiện đại có thể làm tăng thời gian khôi phục sự cố.
- ❖ Trung tâm dữ liệu
    - Sự cố đường truyền và khôi phục nhanh: Trong trung tâm dữ liệu, quy mô mạng lớn và lưu lượng dữ liệu dày đặc khiến sự cố liên kết (link failure) rất phổ biến. Các nghiên cứu cho thấy hỏng kết nối đường truyền là loại lỗi thường gặp nhất và có thể dẫn đến gián đoạn dịch vụ nghiêm trọng, ảnh hưởng trải nghiệm người dùng và thiệt hại tài chính lớn. Do đó, SDN trong Data Center đòi hỏi cơ chế khôi phục tự động và nhanh chóng.
    - Quy mô luồng và tài nguyên phần cứng: Môi trường trung tâm dữ liệu phải xử lý hàng triệu luồng cùng lúc, do đó mà bộ điều khiển phải cài đặt và duy trì bảng quy tắc liên tục. Giới hạn bộ nhớ trên switch có thể bị quá tải khi nhiều quy tắc backup được cài đặt để dự phòng lỗi, làm giảm hiệu suất chuyển tiếp gói và dẫn đến rủi ro tràn bảng. Điều này đòi hỏi thiết kế tỉ mỉ chính sách dự phòng như chỉ cài đặt backup cho luồng

quan trọng để cân bằng giữa tốc độ khôi phục và khả năng xử lý của phần cứng.

❖ Trong môi trường ISP:

- Yêu cầu độ tin cậy cấp nhà mạng: ISP vận hành hạ tầng mạng với yêu cầu độ tin cậy cực cao, gần như không chấp nhận gián đoạn. Hiện nay, các giải pháp chịu lỗi và dự phòng cho SDN còn ở giai đoạn sơ khởi và chưa đủ để đáp ứng tiêu chuẩn nhà mạng. Điều này có nghĩa là các đòi hỏi về bảo mật, đánh giá nguy cơ thất bại phải được nghiên cứu và giải quyết thêm.
- Quy mô rộng và đa miền: Mạng ISP có quy mô lớn và đa miền, khiến SDN đối mặt với thách thức về độ trễ và phân mảnh điều khiển. Khi gặp sự cố, các controller sẽ bị mất đồng bộ và có thể gây trạng thái mâu thuẫn, làm chậm phản ứng. Ngoài ra, việc tích hợp SDN với các giao thức truyền thống như BGP, MPLS dễ phát sinh lỗi nếu không xử lý khéo, ảnh hưởng đến độ ổn định và tin cậy của mạng.

❖ Trong điện toán đám mây

- Mạng ảo hoá phức tạp: Cấu trúc ảo hoá nhiều lớp làm tăng đáng kể độ phức tạp khi giám sát và khắc phục sự cố, hoặc sự tương tác giữa lớp ảo, ví dụ VXLAN, phần mềm khách hàng và lớp hạ tầng vật lý đôi khi giúp phát hiện lỗi nhưng cũng có thể che giấu hoặc gây khó khăn trong việc định vị nguyên nhân gốc rễ. Đây là thách thức lớn trong việc đảm bảo độ tin cậy mạng đám mây, vì lỗi ở bất kỳ lớp nào cũng có thể lan truyền qua các lớp còn lại hoặc khó tái tạo.
- Tính động và đa vùng: SDN controller phải liên tục cập nhật luồng và cấu hình mạng theo tốc độ cực nhanh khi môi trường đám mây thường xuyên thay đổi. Nếu bộ điều khiển hoặc hệ thống điều phối không kịp đáp ứng, mạng có thể trở nên không đồng bộ và phát sinh lỗi. Hơn nữa,

trong kiến trúc đám mây đa vùng, mất kết nối giữa controller và các vùng khác nhau cũng có thể gây ngắt quãng dịch vụ.

3.1.5 Khả năng tương thích với hạ tầng cũ

Một rào cản khác trong việc ứng dụng SDN là khả năng tương thích với hệ thống hạ tầng mạng truyền thống đã và đang được sử dụng trong nhiều năm. Phần lớn các tổ chức, từ doanh nghiệp nhỏ đến các nhà cung cấp dịch vụ quy mô lớn đều đang vận hành trên nền tảng mạng vật lý cũ với thiết bị đa dạng, cấu hình phức tạp và phụ thuộc sâu vào các giao thức định tuyến truyền thống. Việc thay thế toàn bộ hạ tầng hiện tại bằng SDN không chỉ tốn kém mà còn tiềm ẩn nhiều rủi ro cho sự ổn định của hệ thống. Ở các môi trường khác nhau như doanh nghiệp, trung tâm dữ liệu, ISP hay đám mây, mức độ phức tạp và đặc thù trong việc tích hợp SDN với hạ tầng cũ là không giống nhau.

❖ Môi trường doanh nghiệp

- Kiểm soát hạn chế trên đoạn mạng cũ: Trong môi trường lai, SDN controller không thể rà soát hết các nút mạng cũ vì chúng không tham gia cơ chế khám phá mô hình. Do đó mà controller thiếu thông tin tổng thể về mạng. Mặt khác, các định tuyến cũ chỉ cho phép các đường ngắn nhất đơn giản, trong khi SDN mong muốn định tuyến linh hoạt, dẫn đến khó khăn khi kết hợp hai mô hình này. (Hong et al., 2016)
- Triển khai mô hình lai phức tạp: Chỉ khi một phần mạng có SDN, thì các chính sách chi tiết của SDN mới có thể áp dụng trên các switch mới. Các đường truyền nằm hoàn toàn trong hạ tầng cũ vẫn phải tuân theo luồng định tuyến truyền thống, khiến controller không thể áp đặt chính sách lên toàn mạng. Điều này làm giảm khả năng kiểm soát chính sách đồng nhất trên toàn hệ thống.

❖ Trung tâm dữ liệu

- Thiết bị chuyên dụng và chức năng ảo hoá: Các thành phần như bộ cân bằng tải, tường lửa hay thiết bị L4-L7 truyền thống đôi khi không có API mở để SDN điều khiển. Đồng thời, trung tâm dữ liệu thường dùng Open vSwitch, Cisco ACI, hoặc VMware NSX để kết hợp với phần mềm điều phối OpenStack, vCenter. Kết nối giữa controller và các chức năng ảo hóa mạng hoặc thiết bị truyền thống này thường yêu cầu viết cầu nối riêng, tăng độ phức tạp triển khai.
- Kiến trúc phân tầng phức tạp: Môi trường trung tâm dữ liệu thường theo spine-leaf, với nhiều lớp switches. Khi tích hợp SDN, cần đồng bộ định tuyến và bảng nhãn giữa các lớp cũ và lớp SDN để tránh vòng lặp hoặc mất kết nối. Việc này tương tự như trong mạng doanh nghiệp đa tầng, đòi hỏi phải thiết kế kĩ càng từng lớp chuyển mạch và định tuyến sao cho SDN controller có thể phối hợp với các thiết bị legacy trong mọi tầng. (Hong et al., 2016)

❖ Trong môi trường ISP

- Quy mô và giao thức định tuyến chuyên biệt: Mạng ISP có quy mô lớn hơn nhiều so với mạng doanh nghiệp và sử dụng rộng rãi các giao thức định tuyến như BGP và MPLS để kết nối giữa các AS. Các router lõi và biên thường vận hành theo cấu hình truyền thống, khó tích hợp với điều khiển tập trung của SDN. Theo phân tích cả mạng doanh nghiệp và ISP đều có cấu trúc đa tầng với OSPF trong nội bộ và BGP, MPLS giữa các AS. Khi triển khai SDN, cần duy trì hoạt động ổn định của các giao thức cũ song song với điều khiển mới, làm tăng độ phức tạp của mô hình lai (Hong et al., 2016).
- Yêu cầu độ ổn định và phân vùng: Mạng ISP cần độ tin cậy rất cao và khả năng chia phân vùng để phục vụ nhiều khách hàng. Những tính năng

này đã được định nghĩa và tối ưu trên mạng legacy và phải được tái tạo tương đương trên SDN để thực hiện việc ánh xạ các miền phân vùng sang controller, đồng thời giữ tương thích với các giao thức cũ, là một thách thức lớn trong triển khai dịch vụ lai.

❖ Trong điện toán đám mây

- Tương thích công cụ quản lý hiện có: Các nền tảng ảo hóa và dịch vụ đám mây có giao diện lập trình riêng để quản lý mạng. Khi triển khai SDN, controller phải tích hợp với các API này và đồng bộ hóa cấu hình. Các hệ thống quản lý cũ có thể không hỗ trợ trực tiếp các lệnh từ SDN controller, nên cần viết module kết nối hoặc adapter, làm tăng công việc tích hợp.
- Tích hợp mạng ảo hoá và đa thuê bao: Môi trường đám mây dựa trên mạng ảo để phân tách khách hàng trên hạ tầng dùng chung. Các overlay SDN phải kết nối liền mạch với mạng vật lý truyền thống hoặc thiết bị khách hàng. Mặc dù SDN hướng đến khả năng tương thích ngược với hạ tầng hiện có, nhưng trong thực tế phải thiết lập cơ chế dịch giữa mạng ảo và VLAN hoặc MAC cũ. Ví dụ, khi di chuyển dịch vụ từ mạng VLAN truyền thống sang môi trường đám mây dùng VXLAN, cần gateway hoặc NAT đặc biệt để giữ chính sách an ninh, gây phức tạp.

3.1.6 Giao diện cấp thấp

Giao diện cấp thấp như OpenFlow là cầu nối giữa control plane và thiết bị vật lý trong SDN. Tuy nhiên, việc triển khai thực tế gặp nhiều khó khăn do sự khác biệt phần cứng, giới hạn bộ nhớ bảng luồng và độ phức tạp khi ánh xạ lệnh điều khiển. Những khó khăn này thể hiện rõ rệt tùy theo từng môi trường như doanh nghiệp, trung tâm dữ liệu, ISP và đám mây, mỗi nơi có yêu cầu khác nhau về hiệu năng, quy mô và tính linh hoạt.

❖ Môi trường doanh nghiệp

Trong môi trường mạng doanh nghiệp gặp vấn đề về việc sử dụng thiết bị mạng đa dạng, dẫn đến việc hỗ trợ không đồng nhất. Nhiều doanh nghiệp vẫn còn sử dụng thiết bị cũ hoặc sử dụng từ nhiều hãng khác nhau. Thiết bị cũ thường không hỗ trợ các phiên bản OpenFlow mới. Nếu có hỗ trợ, các hãng có thể triển khai các lệnh khác nhau. Điều này buộc bộ điều khiển phải xử lý nhiều trường hợp thiết bị khác nhau, tăng độ phức tạp cho giao diện cấp thấp.

❖ Trung tâm dữ liệu

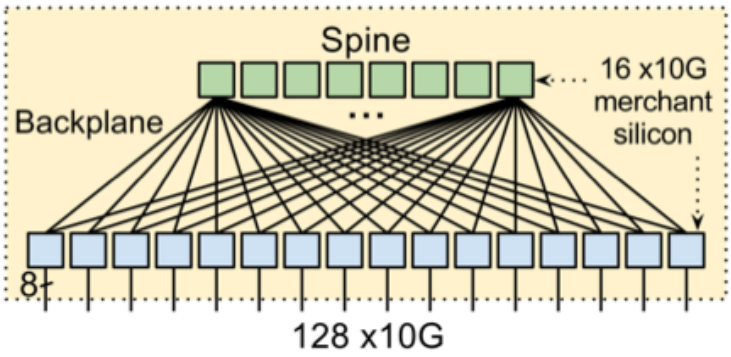
- Tải luồng cao và bộ nhớ hạn chế: Trung tâm dữ liệu tạo ra lưu lượng rất lớn, tuy nhiên các switch SDN có bộ nhớ bảng luồng hạn chế. Khi nhiều luồng mới xuất hiện liên tục, controller phải xử lý và đẩy các quy tắc xuống đường truyền một cách nhanh chóng, dẫn đến overhead giao tiếp cao và có thể trở thành nút cổ chai.
- Độ phức tạp phần cứng và ánh xạ lệnh: Các switch hiện đại có khả năng xử lý đa bảng liên kết phức tạp. Bộ điều khiển phải chuyển đổi các lệnh OpenFlow trừu tượng thành các cấu hình cụ thể cho từng bảng của phần cứng. Ví dụ: “Ưu tiên băng thông cho cổng A” nhưng để thực thi thì framework SDN phải dịch lệnh này thành cấu hình cấp thấp để switch có thể hiểu được, như bảng luồng OpenFlow cụ thể. Quá trình này không đơn giản, vì switch chỉ làm việc ở mức phần cứng cơ bản.

❖ Trong môi trường ISP

- Tích hợp với định tuyến truyền thống: Mạng ISP sử dụng các giao thức định tuyến cũ và thiết bị router truyền thống, kết hợp SDN vào để phối hợp song song hai mô hình điều khiển. Ví dụ, Google B4 đã phải dùng một lớp trung gian gọi là Routing Application Proxy để chuyển đổi các cập nhật định tuyến BGP thành các mục nhập OpenFlow tương ứng cho

switch (Jain et al., 2013). Điều này cho thấy giao diện SDN phải tương tác với hệ thống cũ, làm tăng độ phức tạp của control plane.

- Switch nhiều chipset: Để đạt băng thông cao, switch trong ISP thường ghép nhiều chip silicon, ví dụ B4 dùng 24 chip cho 128 cổng (minh hoạ theo hình 3.5). Controller phải phối hợp bảng luồng của từng chip vật lý và duy trì nhận thức như một switch duy nhất. Việc này làm tăng thêm độ phức tạp của giao diện cấp thấp khi điều khiển các thiết bị switch lớn.



Hình 3.5: Cấu trúc thiết kế chip của switch

(Jain et al., 2013)

❖ Trong điện toán đám mây

Trong môi trường đám mây, các VM thường xuyên được tạo mới, xóa hoặc di chuyển, khiến controller phải liên tục cập nhật bảng luồng cho các switch. Sự tách biệt giữa control plane và data plane gây ra chi phí giao tiếp đáng kể, đặc biệt khi số lượng yêu cầu tăng nhanh trong thời gian ngắn. Điều này dẫn đến độ trễ cao và giảm hiệu năng tổng thể của mạng ở quy mô lớn.

3.2 Giải pháp cho các thách thức của SDN

Để có thể giải quyết được những thách thức mà SDN đang gặp phải như về hiệu suất, khả năng mở rộng, bảo mật, độ tin cậy, tương thích hạ tầng cũ và giao diện lập trình. Các giải pháp dưới đây được đề xuất để có thể tối ưu hoá kiến trúc mạng, từ thiết lập chủ

động giảm độ trễ, kiến trúc phân tán mở rộng quy mô, đến mã hoá OpenFlow tăng bảo mật và API chuẩn hoá để tích hợp dễ dàng.

❖ Hiệu suất và độ trễ:

- Thiết lập chủ động: Là phương pháp cài đặt trước các quy tắc luồng trên các switch trước khi lưu lượng dữ liệu được truyền. Thay vì chờ switch gửi yêu cầu đến bộ điều khiển để xử lý từng gói tin, bộ điều khiển SDN chủ động đẩy các quy tắc xuống switch dựa trên dự đoán về mô hình lưu lượng. Phương pháp này giúp giảm đáng kể độ trễ vì switch có thể xử lý gói tin ngay lập tức mà không cần liên hệ với bộ điều khiển. Ngoài ra, nó giảm tải cho bộ điều khiển, tăng hiệu suất tổng thể của hệ thống.
- Tối ưu hoá bộ điều khiển: Tối ưu hóa bộ điều khiển SDN cải thiện tốc độ xử lý yêu cầu bằng thuật toán hiệu quả, mã nguồn tối ưu (như C++) và phân tải. Điều này làm giảm độ trễ gói tin đầu tiên, tăng khả năng xử lý lưu lượng lớn, nhưng đòi hỏi chuyên môn cao.
- Kiến trúc McNettle: McNettle là kiến trúc SDN tận dụng sức mạnh của hệ thống đa lõi và xử lý song song. McNettle phân chia công việc xử lý giữa các lõi CPU, cho phép xử lý đồng thời nhiều sự kiện mạng, từ đó cải thiện đáng kể hiệu suất và giảm độ trễ. Phù hợp với mạng lưu lượng cao, nhưng cần phần cứng mạnh và phần mềm tối ưu, tăng chi phí.

❖ Khả năng mở rộng

- Kiến trúc phân tán: Kiến trúc phân tán trong SDN sử dụng nhiều bộ điều khiển thay vì một bộ điều khiển trung tâm duy nhất để quản lý mạng. Các bộ điều khiển này phối hợp với nhau để chia sẻ tải xử lý, giúp hệ thống có thể mở rộng để hỗ trợ mạng với hàng nghìn switch và thiết bị. Giải pháp này mang lại khả năng giảm nguy cơ nghẽn cổ chai và tăng khả năng hỗ trợ các mạng quy mô lớn như mạng đám mây hoặc doanh nghiệp toàn cầu.

- Tối ưu hoá thiết lập luồng: Tối ưu thiết lập luồng dùng quy tắc tổng quát, xóa quy tắc cũ, dự đoán lưu lượng để giảm yêu cầu từ switch. Từ đó làm tăng hiệu suất, mở rộng dễ dàng, nhưng cần phân tích lưu lượng chính xác.
- Di chuyển tài nguyên ảo: SDN hỗ trợ di chuyển máy ảo bằng cách cập nhật nhanh quy tắc luồng, đảm bảo kết nối liền mạch. Điều này làm tăng tính linh hoạt, nhưng cần bộ điều khiển hiệu suất cao để tránh gián đoạn.

❖ Bảo mật

- Xác thực và kiểm soát truy cập: Điều này thường được thực hiện thông qua các giao thức xác thực mạnh mẽ như OAuth, Kerberos, hoặc sử dụng chứng chỉ số (TLS/SSL). Ngoài ra, kiểm soát truy cập dựa trên vai trò (RBAC) đảm bảo chỉ người dùng được phép truy cập mạng. Lợi ích của biện pháp này là ngăn chặn truy cập trái phép, bảo vệ dữ liệu nhạy cảm và tăng độ an toàn cho hệ thống.
- Sử dụng ACL thông minh: Danh sách kiểm soát truy cập (ACL) là các quy tắc linh hoạt và tự động được sử dụng để quản lý dung lượng mạng dựa trên chính sách bảo mật. Trong SDN, ACL thông minh có thể được tích hợp với bộ điều khiển để tự động cập nhật dựa trên các thay đổi trong mạng, chẳng hạn như phát hiện mối đe doạ hoặc thay đổi chính sách. Giải pháp này giúp khả năng bảo vệ mạng trước các mối đe doạ mới tăng cao và làm giảm công việc quản trị thủ công.
- Mã hoá kênh OpenFlow: OpenFlow là giao thức chính để giao tiếp giữa bộ điều khiển SDN và switch, và mã hóa kênh OpenFlow đảm bảo rằng dữ liệu truyền qua kênh này được bảo vệ khỏi các mối đe dọa như nghe lén hoặc giả mạo. Điều này thường được thực hiện bằng cách sử dụng giao thức TLS/SSL để mã hóa toàn bộ lưu lượng OpenFlow. Việc mã

hoá này làm hệ thống SDN được tăng cường bảo mật, đặc biệt trong mạng phân tán hoặc môi trường không an toàn.

❖ Độ tin cậy

- Tăng cường dự phòng: Tăng cường dự phòng là chiến lược thiết kế hệ thống với các thành phần dự phòng để đảm bảo hoạt động liên tục ngay cả khi xảy ra sự cố. Trong SDN, điều này có thể bao gồm việc sử dụng nhiều đường dẫn mạng để đảm bảo lưu lượng có thể được định tuyến lại nếu một đường dẫn thất bại, hoặc triển khai các switch dự phòng để thay thế khi phần cứng gặp lỗi. Lợi ích mà giải pháp mang lại là làm giảm thời gian chết (downtime), tăng chịu lỗi. Tuy nhiên, cần đầu tư thêm phần cứng, tăng chi phí.
- Bộ điều khiển dự phòng: Bộ điều khiển dự phòng thay thế khi bộ điều khiển chính thất bại. Trạng thái được đồng bộ với bộ điều khiển chính thông qua các cơ chế sao lưu định kì để đảm bảo liên tục. Khi bộ điều khiển chính thất bại, cơ chế chuyển đổi nhanh sẽ kích hoạt bộ điều khiển dự phòng để tiếp quản. Cách này nâng cao tính ổn định mạng, tuy nhiên việc đồng bộ trạng thái phức tạp và tốn tài nguyên.
- Kiến trúc phân tán: Kiến trúc phân tán không chỉ hỗ trợ khả năng mở rộng mà còn tăng độ tin cậy bằng cách phân chia trách nhiệm quản lý mạng giữa nhiều bộ điều khiển. Nếu một bộ điều khiển hoặc một khu vực mạng gặp sự cố, các bộ điều khiển khác trong hệ thống vẫn có thể tiếp tục hoạt động, đảm bảo tính liên tục của dịch vụ.

❖ Khả năng tương thích với hạ tầng cũ

- Áp dụng mô hình lai: Mô hình lai kết hợp SDN với các mạng truyền thống để tận dụng hạ tầng hiện có trong khi triển khai các tính năng hiện đại của SDN. Điều này thường được thực hiện bằng cách sử dụng các switch lai có khả năng hỗ trợ cả giao thức SDN (như OpenFlow) và các

giao thức truyền thống (như OSPF, BGP). Do đó mà bộ điều khiển SDN có thể quản lý cả các thành phần SDN và truyền thống, tạo môi trường mạng tích hợp.

- Nâng cấp các switch: Nâng cấp các switch hiện có để hỗ trợ SDN, chẳng hạn như cài đặt phần mềm hỗ trợ OpenFlow hoặc thay thế phần cứng tương thích, là một cách để tích hợp SDN vào hạ tầng cũ. Phương pháp này cho phép tổ chức tận dụng các thiết bị hiện có mà vẫn triển khai các tính năng SDN, từ đó giảm chi phí đầu tư mới.
- Tích hợp giao thức truyền thống: SDN có thể tích hợp với các giao thức mạng truyền thống như VLAN, STP hoặc MPLS để đảm bảo tương thích với các thiết bị không hỗ trợ SDN. Bộ điều khiển có thể ảnh xạ các quy tắc SDN vào các cấu hình truyền thống, cho phép quản lý thống nhất cả hai loại mạng.

❖ Giao diện cấp thấp

- Ngôn ngữ lập trình bậc cao: Sử dụng các ngôn ngữ lập trình bậc cao như Python, P4 hoặc Pyretic để lập trình chính sách mạng giúp đơn giản hóa việc phát triển và quản lý SDN. Thay vì làm việc với các giao diện cấp thấp phức tạp, các ngôn ngữ này sẽ cho phép quản trị viên định nghĩa chính sách mạng một cách trực quan và dễ hiểu, chẳng hạn như quy tắc định tuyến hoặc chính sách QoS.
- Chuẩn hoá giao diện: Chuẩn hóa giao diện, chẳng hạn như các API hoặc giao thức như OpenFlow, tạo điều kiện cho việc giao tiếp giữa bộ điều khiển, switch và các ứng dụng mạng. Các giao diện chuẩn hóa đảm bảo rằng các thiết bị từ các nhà cung cấp khác nhau có thể hoạt động cùng nhau, giảm sự phụ thuộc vào một nhà cung cấp cụ thể.
- Đơn giản hoá lập trình chính sách: Đơn giản hóa lập trình chính sách mạng liên quan đến việc cung cấp các công cụ và giao diện giúp quản trị

viên dễ dàng định nghĩa và triển khai các chính sách như QoS, bảo mật hoặc định tuyến. Điều này có thể được thực hiện thông qua GUI, các nền tảng SDN như ONOS hoặc OpenDaylight, hoặc các công cụ trừu tượng hóa chính sách.

Tổng thể, các giải pháp được đề xuất không chỉ giải quyết các thách thức hiện tại của SDN, mà còn hướng đến việc xây dựng một hạ tầng linh hoạt, dễ quản lý và tương thích với hệ thống truyền thống.

CHƯƠNG 4 – KẾT LUẬN

4.1 Đánh giá tổng quan đề tài

Software-Defined Networking là một bước tiến quan trọng trong lĩnh vực quản lý và điều khiển mạng, mở ra hướng tiếp cận linh hoạt, tập trung và hiệu quả hơn so với kiến trúc mạng truyền thống. Với cấu trúc tách biệt giữa mặt điều khiển và mặt dữ liệu, SDN mang lại tiềm năng lớn trong việc tối ưu tài nguyên, tự động hóa quản lý và tăng cường khả năng thích ứng của hệ thống mạng hiện đại.

Tuy nhiên, để phát huy hết tiềm năng đó, SDN phải đối mặt và vượt qua nhiều thách thức quan trọng như độ trễ, khả năng mở rộng, bảo mật, tính tin cậy, khả năng tương thích với hạ tầng hiện có, cũng như sự phức tạp trong thiết kế giao diện và quản lý. Những vấn đề này không chỉ liên quan đến mặt kỹ thuật mà còn đòi hỏi sự thay đổi trong tư duy và quy trình vận hành mạng.

SDN không chỉ đơn thuần là một giải pháp kỹ thuật – mà còn là một cuộc cách mạng trong cách chúng ta xây dựng và vận hành hạ tầng mạng. Do đó, để SDN có thể trở thành nền tảng chủ đạo trong các hệ thống mạng tương lai, cần có sự chuẩn bị toàn diện cả về công nghệ, con người lẫn mô hình triển khai.

4.2 Thuận lợi

Đề tài về SDN hiện đang nhận được sự quan tâm rộng rãi trong cộng đồng nghiên cứu và công nghiệp, do đó có rất nhiều tài liệu học thuật, báo cáo nghiên cứu và các thử nghiệm thực tế được công bố. Việc này giúp quá trình tìm hiểu, phân tích và tổng hợp thông tin cho đề tài "Thách thức trong hiện thực SDN" này trở nên thuận tiện và đáng tin cậy hơn. Bên cạnh việc tập trung vào các thách thức, các nghiên cứu hiện nay cũng mở rộng sang các hướng cải tiến như kiến trúc SDN phân tán, SDN kết hợp với AI, và mô hình lai giữa SDN và mạng truyền thống. Điều này cũng tạo điều kiện để tiếp cận đề tài không chỉ ở khía cạnh hạn chế mà còn ở tiềm năng phát triển trong tương lai.

4.3 Khó khăn

Bên cạnh những thuận lợi khi thực hiện đề tài này, thì cũng không tránh được những khó khăn trong quá trình tìm hiểu. Một trong những khó khăn lớn là số lượng tài liệu tiếng Việt về chủ đề SDN còn khá hạn chế. Phần lớn tài liệu chất lượng cao đều thực hiện và nghiên cứu ở nước ngoài, do đó đòi hỏi phải có khả năng đọc hiểu tốt để khai thác thông tin hiệu quả nhất. Ngoài ra, mặc dù có nhiều tài liệu về lý thuyết và mô hình SDN, nhưng số lượng biểu đồ, đồ thị hoặc số liệu định lượng cụ thể để minh hoạ cho các thách thức như độ trễ, độ tin cậy hay khả năng mở rộng vẫn còn tương đối ít. Điều nay gây khó khăn trong việc trình bày và chứng minh trực quan cho khác vấn đề mà SDN đang đối mặt.

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] Nguyễn Thị Thảo, Trần Vũ Hà, & Lương Minh Quân. (2022, August 15). NGHIÊN CỨU VỀ AN NINH CÔNG NGHỆ MẠNG ĐỊNH NGHĨA BẰNG PHẦN MỀM SDN VÀ ỨNG DỤNG. *Tạp chí Khoa học Nông nghiệp Việt Nam*.

[2] Tô Đức Thiên. (2015, Mar 14). *Software Defined Networking - The Future of Internet*. https://viblo.asia/p/software-defined-networking-the-future-of-internet-1qm6RWeXGeJE

Tiếng Anh

[3] Ali, M., Jehangiri, A. I., Alramli, O. I., Ahmad, Z., Ghoniem, R. M., & Ala'anzy, M. A. (2023, Mar 24). Performance and Scalability Analysis of SDN-Based Large-Scale Wi-Fi Networks. *MDPI*.

[4] Alibaba Cloud. (2024, May 20). *Case Studies: SDN in Action*.

[5] Andersen, G., & MoldStud Research Team. (2024, Jan 24). Exploring Software-Defined Networking (SDN) and its Impact.

[6] Anina Ot. (2022, Apr 14). The Software-Defined Networking (SDN) Market in 2022.

[7] Braun, W., & Menth, M. (2014, May 12). Software-Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choices. *Future Internet 2014, vol 6(2), 302-336*.

[8] Cabaj, K., & Mazurczyk, W. (2016, Aug 24). Using Software-Defined Networking for Ransomware Mitigation: The Case of CryptoWall. *IEEE Network, vol. 30, 2016, pp. 14-20*.

[9] Chickowski, E. (2018, May 25). *Security Must Adjust as SDN Goes Mainstream*.

[10] Cisco. (n.d.). *Software-Defined Networking (SDN) Definition*. Retrieved May 6, 2025.

[11] Dabbagh, M., Hamdaoui, B., Guizani, M., & Rayes, A. (2015). Software-Defined Networking Security: Pros and Cons. *IEEE Communications Magazine*.

[12] He, M., Basta, A., Blenk, A., & Kellerer, W. (2017, May). Modeling Flow Setup Time for Controller Placement in SDN: Evaluation for Dynamic Flows. *IEEE International Conference on Communications*.

[13] Hong, D. K., Ma, Y., Banerjee, S., & Mao, Z. M. (2016). Incremental Deployment of SDN in Hybrid Enterprise and ISP Networks.

[14] Jain, S., Kumar, A., Mandal, S., & Ong, J. (2013, Agu). B4: Experience with a Globally-Deployed Software Defined WAN. *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM Vol. 43.*

[15] Kafazov, A. (2024, Feb 9). *SDN scalability improvements*.

[16] Modali, S. (2016, April 5). *Scaling OpenFlow deployments*.

[17] Mohsin, M. (2022, Oct 9). *SDN Benefits. Simplified.*

[18] Myerson, J. (2018, Aug 24). *What risks does the OpenFlow protocol vulnerability present?*

[19] No Jitter. (2015, Apr 15). *4 Challenges Lying in the Wait of SDN*.

[20] NSA. (2023, Dec). Managing Risk from Software Defined Networking Controllers.

[21] NSA. (2023, Dec 12). NSA Issues Recommendations to Protect Software Defined Networking Controllers.

[22] Open Networking Foundation. (2014, Jun). SDN Architecture Overview.

[23] ONF Solution Brief (2013). SDN Security considerations in the data center. Mike McBride, Editor.

[24] Security Guidance Working Group. (2021, Jun 25). *Cloud Network Virtualization: Benefits of SDN over VLAN*.

[25] Sheehan, J. (2024, Dec 7). *Top Benefits of Software-Defined Networking (SDN)*. SynchroNet.

[26] Stiliadis, D. (2014, Oct 28). *Adventures in Openstack Neutron: Performance and Scale*.

[27] Tellabs. (2014, Feb 11). New Study Reveals: SDN Could Save Operators $9 Billion Globally in Operating Expenses by 2017.

[28] Tootoonchian, A., Gorbunov, S., Ganjali, Y., Casado, M., & Sherwood, R. (2012). On Controller Performance in Software-Defined Networks.

[29] Zhou, Q., & Tan, S.-H. (2024). Latency reduction techniques in SDN-based wide area networks. *Int J Circuit Comput Networking Vol. 5, Issue 1, Part A*, pp 18-23.

[30] Zurier, S. (2015, Aug 3). Governments Find Cost Benefits, Efficiencies in SDN.